

NASA Technical Memorandum 85648

NASA-TM-85648 19850022696

COMPUTER PROGRAM FOR PLOTTING AND FAIRING WIND-TUNNEL DATA

Harry L. Morgan, Jr.

July 1983

~~FOR OFFICIAL USE ONLY~~

Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations.

Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.

Review for general release July 31, 1984

LIBRARY COPY

SEP 6 1983

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

SUMMARY

This report contains a detailed description of the Langley computer program PLOTWD which plots and fairs experimental wind-tunnel data. The program was written for use primarily on the Langley CDC computer and CALCOMP plotters. The fundamental operating features of the program are that the input data are read and written to a random-access file for use during program execution, that the data for a selected run can be sorted and edited to delete duplicate points, and that the data can be plotted and faired using tension splines, least-squares polynomial, or least-squares cubic-spline curves. The most noteworthy feature of the program is the simplicity of the user-supplied input requirements. Several subroutines are also included that can be used to draw grid lines, zero lines, axis scale values and labels, and legends. A detailed description of the program operational features and each subprogram are presented. The general application of the program is also discussed together with the input and output for two typical plot types. A listing of the program code, user-guide, and output description are presented in appendices. The program has been in use at Langley for several years and has proven to be both easy to use and versatile.

X83-10270

INTRODUCTION

Although the primary function of the existing NASA Langley wind-tunnel complex is the same as that of its earlier predecessor NACA, the capabilities of the existing facilities are far superior to those of its predecessor with regard to the relative speeds obtainable, Reynolds number range, and the types and quality of data that can be taken. The advent of modern strain-gage balances, pressure and acoustic transducers, flow visualization systems, engine simulators, attitude measuring devices, and tunnel control and data acquisition systems have resulted in a tremendous increase in the amount of data generated during a typical test program. The recent "explosion" in mini-computer technology has further enhanced the ability to acquire even more data. Without the availability of both on- and off-line high-speed digital and analog computers, the test engineers' ability to acquire, reduce, and analyze this large volume of data would be totally inadequate.

Following a typical wind-tunnel test program, the test engineer is faced with a large amount of data that must be plotted and faired for analysis and report presentation. During the earlier NACA period, these data were plotted and faired manually which required the services of a rather large supporting staff. This manual operation was slow, very tedious and monotonous, and often very inaccurate. However, since the introduction of high-speed computers with peripheral mechanical and optical plotting devices, manual plotting is no longer necessary or desirable. The Langley computing complex currently consists of some of the advanced CDC computers and CALCOMP plotting devices available. The speed, quality, and accuracy of the plots generated by the CALCOMP devices are generally far superior to those generated manually. The available CALCOMP computer software package is extremely versatile and relatively easy to use.

The purpose of this report is to describe the features and application of a computer program written specifically to plot and fair wind-tunnel generated data. This program has the internal designation "PLOTWD" and was written strictly for use on the Langley CDC computer and CALCOMP plotting systems. The program can, however, be adapted to other computer and plotting systems provided the computer has random-access file capability and the plotters have comparable system software routines. This computer program has been used extensively by personnel in the Langley 4- by 7-Meter and 7- by 10-Foot High-Speed wind tunnels for several years and has proven to be very reliable, flexible, and easy to use. In fact, one of the most noteworthy features of this program is the simplicity of the input requirements.

This report contains a discussion of the overall program capabilities and operational structure. A detailed description is included of each subprogram contained in the program. A listing of the computer code is presented in appendix A. Only the subprograms that were written specifically for this program are listed in the appendix. The external subroutines called that are part of the Langley CALCOMP software package are not listed. A description of the user-supplied input requirements and a description of a sample output are presented in appendixes B and C, respectively.

SYMBOLS

$a_0, a_1, a_2, \dots, a_m$	polynomial coefficients
c_i, d_i, e_i, f_i	coefficients of spline equation
h_i	increment in spline independent variable such as $(x - x_i), (y - y_i),$ or $(t - t_i)$
M	order of polynomial
N	number of points
$P(x)$	polynomial equation
S	sum-of-squares of differences
SP	cubic-spline smoothing parameter
w	weighting factor
t	independent variable
x	x-axis dependent variable
x''	d^2x/dt^2
y	y-axis dependent variable
y''	d^2y/dt^2
$\bar{\sigma}$	spline tension factor
δ	increment
Subscripts:	
o	origin
p	plotting sheet
sf	scale factor
wt	wind-tunnel data

PROGRAM OPERATIONAL FEATURES

The computer program PLOTWD was originally formulated to plot and fair sequentially obtained wind-tunnel data on the large 33-inch CALCOMP plotter paper with either 10 or 20 grid lines per inch. The plots generated could be used either as working figures or, after proper border masking and labeling, as final report figures. The later change in the Langley editorial figure presentation standards due to the enhancement of the CALCOMP software package allows for the acceptance of final report figures that are almost totally generated with computer graphics. As a result, additional subroutines were added to the original program version that can be used to draw grid lines, axis scale values and labels, and a legend containing the run numbers and corresponding plot symbols. These plots can be drawn on the plain white CALCOMP paper and, after additional legend information has been added, are suitable report quality figures.

The data on a figure for a particular run are identified by one of 22 available symbols as illustrated in Table I. Each symbol can be drawn with three basic sizes: (1) small (0.100 in.), (2) medium (0.132 in.), and (3) large (0.168 in.). The data are faired using the standard curve fairing technique which utilizes tension-splines to compute a maximum of 100 enhancement points between each pair of input data points. All enhancement points are plotted as a continuous solid line which intersects but does not pass through the symbol surrounding each input data point. During a typical wind-tunnel test, the engineer will often request that repeat data points be taken to either establish anomalies in performance trends or to provide a routine check value. ^{Look!} (Fitting a spline curve through the input data with the repeat points included will almost always produce curves with unrealistic oscillations, especially between original and repeat data points.) To overcome this problem, a subroutine has been included in the plot program which first sorts

the data so that the data points are in monotonic increasing order and then deletes all but one data point in each cluster of duplicate data points. The deleted points are not used as input to the spline curve fit subroutines, but they are plotted as individual data points with the appropriate symbol.

Additional subroutines are available in the plot program that can compute and plot either least-squares polynomial or least-squares cubic-spline curve fits to the input data. These least-squares curve fits are plotted with a continuous dashed line corresponding to the particular symbol as illustrated in Table I. If the user desires, a group of subroutines are available to draw grid lines, axis scale values and labels, and legends. The axis labels that can be drawn consist of a set of nine standard aerodynamic coefficients as illustrated in Table II. A convenient place has been designated in the subroutine that draws labels where the user can add the coding needed to draw other axis labels. The character sets available in the CALCOMP software package are very versatile and the user should be able to draw almost any imaginable label.

The user-supplied input requirements were formulated to be simple and at the same time provide as much flexibility as possible. This flexibility consists of the ability to plot multiple figures on a single sheet, to plot almost any combination of data variables, and to plot almost any combination of runs on a sheet. The input requirements for the basic type of plot without drawn grid lines, axis scale values and labels, and legends are divided into two parts. The first part, designated as the plotting setup information, identifies the plotting sheet boundaries, the symbol size and starting number, curve fairing and editing options, the data array location of the independent test variable, and the combinations of test variables to be plotted with corresponding axis origin locations and scale factors. The second part, designated as the plotting sheet information, is a set of namelists, each

containing a sheet identification number and a list of the runs to be plotted on the sheets.

The wind-tunnel data to be plotted must be written on a file designated as TAPE1 and attached as part of the job execution procedure. The data must be written on TAPE1 with the following unformatted write statement:

```
WRITE(1) IRUN, ITEST, IPT, (DATA(I), I = 1, IMAX)
```

where IRUN is the run number, ITEST is the test number, IPT is the data point number, DATA is an array containing the wind-tunnel data, and IMAX is the maximum number of elements in the DATA array. This input data requirement should be able to accommodate almost any type of wind-tunnel data that is taken sequentially. After reading the setup information during program execution, a subroutine is called that reads the data on TAPE1 and reloads it on the random-access file TAPE2 with an equivalent run number indexing. Thereafter, anytime a particular run is to be plotted, the corresponding data for that run is read from the random-access file and loaded into a plotting array. Retrieving the data in this manner simply means that the file read of the data for a particular run begins immediately at the first data point rather than each time having to read the data file from the beginning to locate the first point. This random-access feature greatly reduces the total computing costs by reducing the execution time and the number of I/O operations.

DESCRIPTION OF COMPUTER PROGRAM

The wind-tunnel data plotting program PLOTWD consists of a main program, 18 subroutines, and 2 function subprograms. A computer listing of the program coding is presented in Appendix A. A description of the input requirements for the program is presented in Appendix B and a description of the output for a sample input case listed in Table III is presented in Appendix C. The primary function and execution sequence for the main program and each subprogram are discussed in this section. A supplemental list and corresponding description of the input variables and internal parameters for each routine is included.

Program PLOT

The function of program PLOT is to control the overall execution of the plotting process. Listed at the beginning of the program are a group of comment statements that describe the program input requirements. A group of dimension and common statements appear next consisting of the arrays containing the run numbers and plotting variables, axis origin locations, and axis scale and label information. Several global program constants are then defined and calls made to subroutines PSEUDO and LEROY to initialize the plot vector file SAVPLT for subsequent post-process plotting on a variety of plotters at Langley. A call is then made to subroutine SETUP which reads the input plotting setup data. This is followed by a call to subroutine DATADK which loads the input wind-tunnel data that is sequentially written on the input file TAPE1 into the random-access working file TAPE2. This subroutine is called only once during the program execution; therefore, if the plotting setup information is changed during the program execution, the test number may not change.

The data array locations of the independent and dependent variables to be plotted are then loaded into the work array NLOC. A check is then made to determine if any parts of the plotting setup have common abscissas or ordinates and, if so, the corresponding internal XPLT and YPLT array elements are set equal to 1.0. These internal arrays are checked during the axis labeling and zero-line drawing execution step to prevent duplication. The next execution step is the read of the input namelist SHEET which contains the sheet number and corresponding run numbers to be plotted. The test, sheet, and run numbers are then drawn at the bottom of the plotting sheet and the sheet frame drawn. Arrowheads are then drawn along the bottom and left-hand edge of the sheet frame to indicate the origins. The corresponding axis scale and label are then drawn adjacent to each arrowhead. Zero lines which run from the

arrowhead to the opposite frame border are then drawn if the parameter LZERO is equal to zero.

Execution then proceeds to plot the desired wind-tunnel data array variables for each run. The first step is an identification of the plotting symbol to use. The 22 symbols available are presented in Table I in the standard order. The number of the symbol to use is the same as the input order of the run number in the SHEET namelist. If the user desires to skip a particular symbol, simply input a run number of zero at the appropriate place in the run number list in the SHEET namelist. If the user desires to set up a symbol order different from the standard order, this can easily be accomplished by redefining the sequential values in the internal array LSYM.

The next step is to read the particular wind-tunnel data corresponding to the variable to be plotted from the random-access file TAPE2 and to load these data into the internal array VAR. The next step is to convert the wind-tunnel data from coefficient form to the equivalent x_p and y_p locations on the plotting sheet using equations

$$x_p = x_{wt}/x_{sf} - x_o \quad (1)$$

$$y_p = y_{wt}/y_{sf} - y_o, \quad (2)$$

and to load the converted values into arrays X and Y, respectively. At the same time, the values for the independent data variable are loaded into array T and a check is made of the x_p and y_p data to determine if any points are outside the sheet frame. If any points fall outside the sheet frame, a message will be printed stating the run and plot number and the number of points outside the frame. The next step is to sort and edit the x_p and y_p data if the user specified a nonzero value for the input parameter IEDIT. The final step is to call subroutine CURPLT which plots and fair the x_p and

y_p data using tension splines. After plotting all of the specified variable combinations for a particular run, the above sequence of steps is repeated for the next run.

After plotting all the specified runs for a particular sheet, the plotter is advanced to the start location for the next sheet, the next SHEET namelist read, and the entire sequence of scaling and plotting steps are repeated. The next execution step in the program is to print a summary of the sort and edit information. The final execution step is to draw a statement indicating that all plotting has been completed and to call subroutine CALPLT to close the plot vector file SAVPLT. The following is a list and description of the parameters used in this program:

LSYM	array containing integer values corresponding to the symbol order
RUN	array containing run numbers to be plotted
NPT	array containing the directory or index information on the random-access file TAPE2 (dimensional size equivalent to number of data points on input wind-tunnel data file TAPE1)
XTAPE, YTAPE	arrays containing wind-tunnel data array indices of x_{wt} and y_{wt} variables
XOFFSET, YOFFSET	arrays containing origin x_0 and y_0 locations on plotting sheet, in.
XSCALE, YSCALE	arrays containing axis scale factors x_{sf} and y_{sf} , change in coefficient per inch
XLABEL, YLABEL	arrays containing x- and y-axis labels
XPLT, YPLT	arrays containing a value of 1.0 if the particular x- or y-axis is repeated
X,Y	arrays containing scale x_p and y_p values, in.
T	array containing value of independent variable

VAR	two-dimensional array containing y_{wt} data in one level and x_{wt} data in the next level
NLOC	array containing consecutive DATA array indices of y_{wt} and x_{wt} variables
NO	sheet number
NEWCASE	if = 1, new setup deck follows the current SHEET namelist if = 0, new SHEET namelist follows
JREAD	tape number of file containing input setup and namelist data
NRNMAX	maximum number of allowable RUNS per sheet
NPLMAX	maximum number of allowable plots per sheet
NDMAX	maximum number of allowable data points per run
NPMAX	maximum number of total data points on TAPE1
IPRT	if = 0, print sort-and-edit summary data if = 1, do not print sort-and-edit summary data
LZERO	if = 0, draw zero lines if = 1, do not draw zero lines
NORG	if = 0, draw arrowheads at axis origins if = 1, do not draw arrowhead at axis origins
IDISK	if = 0, random-access file TAPE2 has not been loaded if = 1, TAPE2 has been loaded
ITEST	test number
SHEETW, SHEETH	plotting sheet width and height, in.
SPACE	space between plotting sheets, in.
ISYM	starting symbol number
ISIZE	symbol size
IOP	data fairing option
TENSION	spline tension factor σ

IEDIT	data sort-and-edit option
TOLR	editing tolerance of independent variable
IW	DATA array index of independent variable
NPLOT	number of desired plots per sheet
IERR	if = 1 after call to subroutine SETUP, indicates that last case has been read or error occurred if = 1 after call to subroutine DATADK indicates that error occurred during either a read or write of TAPE2
NVAR	number of variables read from TAPE2 for each data point (2*NPLOT+1)
NP	number of data points to be plotted for a specified run

Subroutine SETUP

The function of subroutine SETUP is to read and print the plotting setup information as described in Appendix B. A sample of the print formats are presented as page 1 output in Table IV. As various parameters are read from the input file, checks are performed to insure that the parameter is within the program limitations. If the particular parameter cannot be safely redefined within the program limitations, an error message is printed and program execution terminated. All the parameters in the subroutine argument list are defined in the description of program PLOT.

Subroutine DATADK

The primary function of subroutine DATADK is to read data from or write data to the random-access file TAPE2. If the ICODE parameter equals 0, the wind-tunnel data on the input file TAPE1 is read and then written on the random-access file using the mass storage write statement WRITMS. Each data point is read from TAPE1 using the following read statement:

```
READ(1) IRUN, ITEST, IPT, (DATA(I), I = 1, IMAX)
```

The information in the data array is then shifted forward three array locations and the values of IRUN, ITEST, and IPOINT loaded into the first three locations. The information in the resultant DATA array is then written on the random-access file TAPE2 and the access location loaded into the array NPT. This access location will be needed during all subsequent reads of this file; therefore, the user should not add any additional coding that would destroy the contents of this array. If the data being written on TAPE2 corresponds to that for the first point of each run, the run number and the array index in the NPT array are written on a local file TAPE12 for use during the later reads of TAPE2. The default value for the parameter IMAX is 300. If the input data tape contains either more or less variables for each data point, the value of IMAX must be redefined and the dimension size of the DATA array redefined to a value equal to IMAX + 3. After transferring the data on TAPE1 to TAPE2, a summary of the run numbers available and the total number of data points transferred is printed as illustrated in Table IV as page 2 output.

If the ICODE parameter equals 1, all the wind-tunnel data points for a specified run number IRUN are read from the random-access file TAPE2 and the specified variables defined by the user-supplied input values for XTAPE, YTape, and IW are then loaded into the two-dimensional array VAR. The random-access index location for the first point of the specified run is determined by reading the run and NPT array index information written previously on TAPE12. During the read of each data point on TAPE2, the data are loaded into the array DATA. The subsequent transfer of the information in the DATA array to the VAR array is shifted back three array locations to prevent loading the values for IRUN, ITEST, and IPT into the VAR array. The test number read from TAPE2 is, however, checked against the test number specified by the user in the setup information to ensure that the correct test data were input on TAPE1.

Several types of errors can occur which may cause the program execution to either terminate or proceed to the next specified run number. The types of errors that may occur include (1) there are no data on TAPE1, (2) the total number of input data points exceeds the allowable maximum, (3) the input and specified test numbers are different, (4) the specified run number is not on TAPE2, (5) number of data points for a specified run number exceeds the allowable maximum, (6) the number of data points on TAPE2 for a specified run cannot be determined from the information written on TAPE12, and (7) the specified XTAPE or YTAPE value exceeds the allowable maximum IMAX. If any of these errors occur, an appropriate error message will be printed. Most of the important parameters used in this subroutine are defined in the description of program PLOT. The following additional parameters are used in this subroutine:

DATA	array containing x_{wt} and y_{wt} data read from either TAPE1 or TAPE2 (must be dimensioned by IMAX+3)
IMAX	maximum number of elements in data array
IRUN	specified run number
IPT	input data point number
ICODE	if = 0, read data on TAPE1 and write to random-access file TAPE2 if = 1, read data on TAPE2 for a specified run and load into the VAR array
OPENMS	system subroutine to open and initialize random-access file
READMS	system subroutine to read information from random-access file
WRITMS	system subroutine to write information on random-access file

Subroutine EDIT

The function of subroutine EDIT is to sort and edit a set of x_p and y_p data to be plotted for a specified run. If the user desires to fair

the wind-tunnel data with the tension spline method, the set of data to be faired must be arranged so that the corresponding values for the independent variable, t , are monotonically increasing, single-valued, and reasonably spaced to prevent large oscillations in the faired curve. The first time this subroutine is called with a specified run number, the results of the sort and edit procedure are written on the output file TAPE7 and the run number on output file TAPE11 for future printing during the call to subroutine PRTEDT. Therefore, the first execution step in this subroutine is a check of the information on TAPE11 to determine whether or not summary data have previously been written on TAPE7. If so, the parameter NSTORE is set equal to a value of 1.

The next execution step is to sort the input values of the independent variable, t , so that the final set of values are monotonically increasing. The sorting is accomplished in an iterative manner with the following steps:

- (1) Load remaining unsorted values of the independent variable and their corresponding indices into temporary arrays TEDT and IEDT, respectively.
- (2) Determine the minimum TEDT value and its corresponding index and load index value into array ISAVE.
- (3) Delete the minimum TEDT value from the TEDT array.
- (4) Repeat steps 1, 2, and 3 until all input values of the independent variable have been processed.
- (5) Load sorted independent variable values and corresponding y_p values into temporary arrays using index information in the ISAVE array.

If the user specified on edit tolerance TOLR that is greater than zero, the sorted data will be edited next to delete all but one data point in each cluster of repeat points. Repeat points are defined as any two adjacent points whose difference in the absolute values of the independent variable

is less than a specified tolerance. The edit procedure consists simply of connecting the two data points adjacent to a particular cluster of repeat points with a straight line and then determining the point in the cluster closest to the line. The closest point will be saved and the remaining points in the cluster will be deleted. These deleted points are, however, plotted immediately on the plotting sheet with the appropriate symbol.

Although this editing procedure is simple, four problems may be encountered which will prevent the sorted data from being properly edited. If any of these particular problems occur, the error flag IERR will be set equal to the problem number, all data points input for the particular run plotted with the appropriate symbol, and control returned to the main program PLOT. Program execution will then proceed to plot and fair the next specified run number. The first error which may occur is that all the values for the sorted independent variable are within the specified tolerance TOLR. This error usually occurs because the user specified a tolerance which is too large. The remaining errors occur due to the presence of several adjacent clusters of repeat points which prevents formation of the straight line between adjacent data points. The second flagged error indicates that all data points except the last are clustered; the third flagged error, that all data points prior to the particular point being edited are clustered; and the fourth flagged error, that all data points except the first are clustered. If any of the latter three flagged errors are printed in the output, the user should carefully examine the data and decide which data points to delete.

Following the editing procedure, a summary of the sort and edit information is written on TAPE7 provided the value of the NSTORE parameter is zero. The last execution step is to reload the independent variable t , x_p , and y_p input arrays with the sorted and edited values. The following additional parameters are used in this subroutine:

KEDT	number of run numbers written on TAPE11
TIN	array containing input values of independent variable t
TEDT	array containing sorted and edited values of independent variable
TSAVE, YSAVE	temporary arrays containing values of t and y_p
IEDT, ITEMP,	temporary arrays containing indices of sorted or edited t
ISAVE	values
TMIN	minimum value of t
NSTORE	if = 0, summary of sort and edit information for particular run is to be written on TAPE7
NSYM	plot symbol number for deleted points

Subroutine PRTEDT

The function of subroutine PRTEDT is to read the sort and edit information generated during the calls to subroutine EDIT. This information is read from TAPE7 and consists of the input, sorted, edited, and deleted values of the independent variable t for all of the runs requested during the job execution. A sample of the output is illustrated as page 4 output in Table IV. As previously noted, the sort information is the same regardless of the number of times the particular run is plotted. However, the edit information is only applicable to the first plot of the run. All of the important parameters used in this subroutine are defined in the description of subroutine EDIT.

Subroutine ZEROLN

The function of subroutine ZEROLN is to draw heavy lines on the plot perpendicular to the axis at each XOFFSET and YOFFSET location to indicate origins or zero lines for the variables to be plotted. Each heavy line is generated by simply drawing a straight line perpendicular to the sheet border at the specified origin location and then drawing two additional lines

parallel to the original line and offset to either side of it. The following subroutine arguments and parameters are used:

XO, YO	starting x_p and y_p plot locations of the zero line, in.
DIST	length of zero line, in.
LINE	if = 1, zero line parallel to x-axis if = 2, zero line parallel to y-axis
D	offset distance of additional lines drawn parallel to zero line (width of zero line = $2 \times D + \text{plotting pen width}$), in.

Subroutine CURPLT

The function of subroutine CURPLT is to plot and fair a tension spline curve through an input set of x_p and y_p values. The subroutine contains a plot option to plot the symbols only (IOP = 0) and a plot option to plot the symbols and draw a faired curve between them (IOP = 1). For both plot options, the symbols are plotted by calls to subroutine PNTPLT which has been attached as part of the CALCOMP software package. If the curve fairing option has been chosen, the input data points x_p and y_p are faired with a tension spline that has a tension factor equal to the value of the input parameter TENSION. The coordinates of the faired curve are determined by fitting a tension spline through both the x_p and y_p as a function of the independent variable t and then interpolating the two curves to obtain the resultant y_p as a function of x_p faired curve which is then drawn on the plot. If the x_p is a scaler function of t (XTAPE value equals IW), the double interpolation process is not necessary. The input value of the parameter IVAR must be equal to 0 if x_p and t are equal and equal to 1 if they are not equal. The first execution step is a check to ensure that the independent variable values are monotonically increasing. The next step is a call to subroutine CUBSPL which determines the value of the second derivative x_p'' and y_p'' of the tension spline at the input points. The final execution step is to

interpolate a specified number of spline coordinates between each pair of input points and to draw the curve so that it does not cross the borders of the symbol surrounding each data point. This capability represents a unique feature of this program.

The following discussion outlines the procedure used to determine the intersection points of the spline curve with the border of each end point symbol and the number of coordinates to interpolate between the pair of points. The straight line distance between the pair of points is computed first. If the length of the line is less than the nominal width of the symbol, the spline curve cannot be drawn between the pair of symbols and execution advances to the next pair of points. If the length of the line is greater than the nominal width of the symbol, the interpolation increment of the independent variable, t , between the pair of points is determined using the formula

used to determine if enough room between symbols to generate firing.

$$\delta = \frac{t_{i+1} - t_i}{(NH-1) I(t)} \quad (3)$$

where $NH = 101$ and $I(t)$ is the integer portion of $(t_{i+1} - t_i)$. Spline coordinates are then computed at consecutive δ increments between the pair of points.

The next step is to determine the intersection points between the spline curve and the borders of the symbols drawn at each point. This step is accomplished utilizing an iterative technique that is centered around the capabilities of subroutine SYMBOL which determines the coordinates of the border of a specified symbol as a function of the angular position of a radial line drawn through the symbol center. The iteration cycle begins at a point on the spline curve from the symbol center corresponding to a value of t equal to an

incremental value, H, based on the symbol size. The coordinates of the spline curve are interpolated using the function subprogram FUNC and then the straight line length and angular location computed from the center of the symbol. The angular value is then input to subroutine SYMBOL which determines the corresponding coordinates on the symbol border. The straight line length from the symbol center to the border is then computed. If the spline straight line length is less than that from the center to the border, the value of t is incremented by H and the above steps are repeated. If the spline straight line length is greater than that from the center to the border, the increment H is halved and added to the previous t value and the above steps repeated. This procedure is repeated until the difference between the spline and symbol-border straight line lengths is within a tolerance equal to the value of the parameter EPS.

After determining the intersection coordinates of the spline curve and the symbol borders for a pair of points, the final step is to plot the spline curve between the intersection coordinates. The LINE subroutine, which is part of the CALCOMP software package, is used to plot this curve. This subroutine draws a series of connecting straight lines between an input set of coordinates. By plotting a relatively large number of closely spaced coordinates, the smoothness of the curve between each pair of input points is retained. The following additional subroutine arguments and parameters are used in this subroutine:

IVAR	if = 0, independent variable t is plotted on x-axis
	if = 1, a dependent variable is plotted on x-axis
IOP	if = 0, plot symbols only
	if = 1, plot symbols and spline curve

MX, MY	arrays containing second derivative values x_p'' and y_p'' of tension spline curve fit
DS1, DS2	arrays containing interpolated x_p and y_p spline curve coordinates between a pair of input points
DUMX, DUMY	arrays containing interpolated x_p and y_p spline coordinates between the borders of the symbols surrounding a pair of input points
NH	number of interpolated spline coordinates between a pair of input points
EPS	tolerance used to determine the intersection point between each symbol border and the spline curve
H	initial increment in t value

Function ATANF

The purpose of function subprogram ATANF is to compute the angular position of a specified coordinate point in degrees with the quadrants defined from 0 to 360 degrees. This function is used with subroutine CURPLT to determine the radial angle of a straight line connecting a specified point on the spline curve and the center of the nearest symbol.

Subroutine SYMBOL

The function of subroutine SYMBOL is to compute the coordinates of the intersection between the border of a specified symbol and a radial line drawn from the center of the symbol. This requires that the border of each symbol presented in Table I be defined as a function of the angular equivalent of border coordinates with the origin of the axis system at the symbol center. The shapes of 22 symbols are defined in this subroutine. The first 10 symbols are the NASA standard open symbols: (1) circle, (2) square, (3) diamond, (4) triangle, (5) right triangle, (6) quadrant, (7) dog house, (8) fan, (9) long diamond, and (10) house. The next 10 symbols are the standard centered

(center of symbol indicated by +) symbols corresponding to the standard open symbols. The 21st symbol is a period and the 22nd symbol is a plus sign. The symbol border for the last two symbols is an imaginary circle with a radius 10 percent larger than the symbol height. The shapes of the 10 basic open and centered symbols are made up of various combinations of circles, squares, and triangles. Most of the symbols are symmetric about the local y-axis except the right triangle and the quadrant. The shape of each symbol is determined by computing the straight line or circular arc coordinates of the various segments that make up the symbol. The equations and angular positions of the segments that make up each of the 10 basic symbols are relatively easy to derive and, therefore, are not presented in this report. The following arguments and parameters are used in this subroutine:

NO	symbol number
IS	symbol size (1 for small, 2 for medium, and 3 for large)
X,Y	x and y coordinates of intersection of symbol border and radial line drawn from the center of the symbol
T	angular position of radial line, deg.
SCALE	relative height of three basic symbol sizes, in.

Subroutine CUBSPL

The function of subroutine CUBSPL is to fit a cubic or tension spline through an input set of x and y values. The cubic spline represents a special case of the tension spline where the tension is zero. The input x and y data are used to compute a matrix of simultaneous equations in the form of

$$e_i y_{i-1}'' + d_i y_i'' + f_i y_{i+1}'' = c_i \quad (4)$$

for $i = 2, 3, \dots, N-1$ where for the cubic spline

$$e_i = \frac{h_{i-1}}{6}, \quad d_i = \frac{h_{i-1} + h_i}{3}, \quad \text{and} \quad f_i = \frac{h_i}{6} \quad (5)$$

and for the tension spline

$$c_i = \frac{1}{\sigma^2} \left[\frac{1}{h_{i-1}} - \frac{\sigma}{\sinh(\sigma h_{i-1})} \right],$$

$$d_i = \frac{1}{\sigma^2} \left[\frac{\sigma \cosh(\sigma h_{i-1})}{\sinh(\sigma h_{i-1})} - \frac{1}{h_{i-1}} + \frac{\sigma \cosh(\sigma h_i)}{\sinh(\sigma h_i)} - \frac{1}{h_i} \right], \quad (6)$$

and $f_i = \frac{1}{\sigma^2} \left[\frac{1}{h_i} - \frac{\sigma}{\sinh(\sigma h_i)} \right]$

For both cubic and tension spline,

$$c_i = \left(\frac{y_{i+1} - y_i}{h_i} \right) - \left(\frac{y_i - y_{i-1}}{h_{i-1}} \right),$$

$$h_i = x_{i+1} - x_i, \quad (7)$$

and $h_{i-1} = x_i - x_{i-1}.$

For the tension spline, the tension parameter σ is the average tension per length of the input x values and determined using the equation

$$\sigma = \frac{\bar{\sigma} (N-1)}{(x_N - x_1)} \quad (8)$$

where $\bar{\sigma}$ is the user-supplied value of the input parameter TENSION and N is the number of input values. A complete derivation and description of cubic and tension splines are presented in reference 1.

The second derivatives are the unknowns in the system of equations generated by equation (4). The matrix of terms is tridiagonal with two less unknowns than equations; therefore, the second derivative at the end points must be specified. In this subroutine, the second derivatives at the end points are computed by fitting a second-order polynomial of the form

$$y = a_1 x^2 + a_2 x + a_3 \quad (9)$$

to each end point and its two adjacent input points and then differentiating

to determine the second derivative which is

$$y'' = 2a_1 \quad . \quad (10)$$

The resultant tridiagonal matrix can be solved using the Crout reduction method which is described in detail in reference 2. Using this method, the solution becomes a simple back substitution

$$y''_N = c_N \quad \text{for } i = N \quad (11)$$

$$\text{and } y''_i = \bar{c}_i - \bar{f}_i y''_{i+1} \quad \text{for } i = N-1, N-2, \dots, 1$$

$$\text{where } \bar{d}_i = d_i - e_i \bar{f}_{i-1} ,$$

$$\bar{f}_i = f_i / \bar{d}_i \quad , \quad (12)$$

and

$$\bar{c}_i = \frac{c_i - e_i \bar{c}_{i-1}}{\bar{d}_i}$$

The following is a description of the parameters in the argument list for this subroutine:

X,Y	input arrays containing table of x- and y-values
N	number of input coordinates
YPP	output array containing y'' values
TENSION	input tension factor $\bar{\sigma}$
TENS	output dimensionalized tension factor σ
A	internal work array used during back-substitution process

Function FUNC

The function of subprogram FUNC is to compute the y-value of a spline curve at a given x-value in a specified interval $x_i < x < x_{i+1}$.

The interpolation equation for the cubic spline is

$$y(x) = y_i'' \left[\frac{(x_{i+1} - x)^3}{6h_i} - \frac{(x_{i+1} - x) h_i}{6} \right] + y_{i+1}'' \left[\frac{(x - x_i)^3}{6h_i} - \frac{(x - x_i) h_i}{6} \right] + \left[\frac{y_i (x_{i+1} - x) + y_{i+1} (x - x_i)}{h_i} \right] \quad (13)$$

and for the tension spline is

$$y(x) = \frac{y_i''}{\sigma^2} \left[\frac{\sinh [\sigma(x_{i+1} - x)]}{\sinh (\sigma h_i)} - \frac{(x_{i+1} - x)}{h_i} \right] + \frac{y_{i+1}''}{\sigma^2} \left[\frac{\sinh [\sigma(x - x_i)]}{\sinh (\sigma h_i)} - \frac{(x - x_i)}{h_i} \right] + \left[\frac{y_i (x_{i+1} - x) + y_{i+1} (x - x_i)}{h_i} \right] \quad (14)$$

where $h_i = (x_{i+1} - x_i)$.

Subroutine DASHLN

The function of subroutine DASHLN is to draw a dashed line spline curve through an input set of x_p and y_p values. The spline curve fit procedure for this subroutine is very similar to that described for subroutine CURPLT with the major difference being that symbols are not drawn around each symbol with this subroutine. Therefore, the complex procedure to determine the intersection between the spline curve and the symbol border is not included in this subroutine. The dashed lines that can be drawn consist of various combinations of long and short dashes as illustrated in table I.

The subroutine CUBSPL is called initially to determine the y_p'' values of a tension or cubic spline fit of the input data. A specified number of equally spaced points are then interpolated between the input end points and the total length of the curve computed using a simple triangular summation method. The

basic length of the long and short dash combination and the space between combinations are then scaled so that the final plotted dashed line will start and end with a complete long and short dash combination. The scaled dashed line is then drawn using linear interpolation of the spline curve to determine the end points x_p and y_p values for each consecutive long or short dash. The following additional subroutine arguments and parameters are used in this subroutine:

NSYM	symbol or dashed line number
IOP	if = 0, plot symbols only if = 1, plot dashed line only
NL	number of long dashes in each combination
NS	number of short dashes in each combination
XI, YI	arrays containing interpolated x_p and y_p values of spline curve
S	array containing interpolated length of spline curve
NP	number of interpolated spline curve values
SL	length of long dash, in.
SS	length of short dash, in.
SP	length of space between dashes, in.
NLT	total number of long and short dash combinations to be plotted
DTN/DT	scaling factor for each dash combination

Subroutine LINEAR

The function of subroutine LINEAR is to determine the x and y values at a specified t value using linear (straight line) interpolation. The first execution step is to determine the interval containing the specified t value. The final step is to compute the corresponding x and y values using the following linear interpolation equations:

$$x = x_{i-1} + \delta (x_i - x_{i-1})$$

$$y = y_{i-1} + \delta (y_i - y_{i-1}) \quad (15)$$

where $\delta = (t - t_{i-1}) / (t_i - t_{i-1})$.

Subroutine LSQPLT

The function of subroutine LSQPLT is to draw a least-squares polynomial or least-squares cubic spline curve through an input set of x and y values. For each set of input data, the input points are plotted with the specified symbol and the least-squares curve plotted with the corresponding dashed line. The curve-fitting procedure used in this subroutine is similar to that used in subroutine CURPLT. A separate least-squares curve is determined for each set of input x and y values as a function of the independent variable t and a y(x) curve then obtained by interpolation. If the user specifies that a least-squares polynomial curve be drawn through the input data, subroutine LSQ is called which determines the coefficients of the specified-order polynomial. If the user specifies that a least-squares cubic-spline curve be drawn, subroutine CSDS is called which determines the coefficients of the third-order piecewise polynomials that constitute the spline curve.

Following the call to the selected least-squares subroutine, the x and y values at each input t value are computed and sum-of-the-squares of the differences between the input and least-squares values are computed. If the user specifies a nonzero value for the parameter IPRINT, a summary of the least-squares curve fit process will be printed as illustrated in table V and VI. The final execution step is a call to subroutine DASHLN which draws the least-squares compute x and y values with the specified dashed line. The following additional arguments and parameters are used in this subroutine:

ILSQ if = 0, fit input data with least-squares polynomial
 if = 1, fit input data with least-squares cubic spline
 NPOL order of polynomial if ILSQ = 0
 DF standard deviation for least-squares cubic spline if
 ILSQ = 1
 COEF two-dimensional array containing coefficients of piecewise
 third-order polynomials of cubic spline
 CX,CY arrays containing coefficients of least-squares polynomial
 XNEW,YNEW arrays containing new x and y values after curve fit
 WT weighting factor used in least-squares polynomial curve fit
 IPRINT if = 0, do not print summary data
 if = 1, print summary data
 ERRX,ERRY sum-of-squares of the differences between input and
 least-squares x and y data

Subroutine LSQ

The function of subroutine LSQ is to determine the coefficients of a polynomial that best fits an input set of weighted x and y data. The polynomial is in the form

$$p(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + \dots + a_M x_i^M \quad (16)$$

where M is the order of the polynomial. The sum-of-the-squares differences between the weighted input and polynomial fit values are

$$S = \sum_{i=1}^N (P(x_i) - w_i y_i)^2 \quad (17)$$

The least or minimum error is

$$\frac{\partial S}{\partial a_j} = 2 \sum_{i=1}^N \left[P(x_i) - w_i y_i \right] x_i^j = 0 \quad (18)$$

where $j = 0, 1, 2, \dots, M$. This reduces to a set of simultaneous equations in the form

$$a_0 \sum x_i^j + a_1 \sum x_i^{j+1} \dots + a_M \sum x_i^{j+M} = \sum w_i y_i x_i^j \quad (19)$$

which can be solved for the polynomial coefficients a_j using a simplified solution technique that takes advantage of the symmetric properties of the matrix of terms generated by the left-hand side of equation (19). The following arguments and parameters are used in this subroutine:

X,Y	arrays containing input x_i and y_i data
W	array containing input weighting factor w_i
NP	number of input points N
N	order of polynomial M
C	array containing values of the coefficients of the least-squares polynomial
A	two-dimensional work array

Subroutine CSDS

The function of subroutine CSDS is to fit a least-squares cubic-spline through an input set of x and y data. The method used in this subroutine defines a continuous cubic-spline function in the form of the third-order polynomial

$$P(x) = a_0 + a_1 h_i + a_2 h_i^2 + a_3 h_i^3 \quad (20)$$

for $i = 1, 2, 3, \dots, N - 1$ where $h_i = (x - x_i)$ and N is the number of input points. The coefficients a_0, a_1, a_2 , and a_3 are computed in a least-squares manner so that

$$\sum_{i=1}^N \left[\frac{P(x_i) - y_i}{\delta y_i} \right] \leq SP \quad (21)$$

$$\text{and } \int_{x_1}^{x_N} \left[\frac{d^2 P(x)}{dx^2} \right] dx \text{ is a minimum} \quad (22)$$

where the smoothing parameter SP is in the interval $N - \sqrt{2N} \leq SP \leq N + \sqrt{2N}$ and δy_i is specified allowable standard deviation in the error in y_i .

A detailed discussion of the least-squares cubic-spline method is presented in reference 3 and, therefore, will not be included in this report. This subroutine is also a part of the standard math-library subprogram package available on the Langley CDC computer system and is identified by the same name and argument list. A complete description of the input and output parameters in the argument list are presented at the beginning of the listing of the subroutine in appendix A.

Subroutine AXISLB

The primary function of subroutine AXISLB is to draw scale values on the x or y axis. The scale value is drawn in one-inch increments starting at the origin and continuing for a specified length. Each scale value drawn is centered adjacent to the inch mark for the y-axis or centered below the inch mark for the x-axis. The following is a description of the subroutine arguments:

IAXIS	if = 1, draw x-axis scale values if = 2, draw y-axis scale values
XORG, YORG	x_p and y_p values of axis origin, in.
ORG	scale value at origin
SCALE	increment in scale value per inch

HT height of scale values, in.

NDIG number of significant figures to draw on the right side of
the decimal point (NDIG = -1 will drop decimal point)

Subroutine COEFSY

The function of subroutine COEFSY is to draw a specified standard aerodynamic label as illustrated in table II. The first seven labels are the standard designations for the lift, drag, pitching-moment, rolling-moment, yawing-moment, and side-force coefficients and lift-drag ratio. The last two labels are the standard designations for the angle of attack and angle of sideslip. The width-height ratio for each label is also listed in table II and is useful to the user when determining the x_p and y_p values needed to position the label. If the user desires to add coding to draw other labels, the additional coding can be inserted following the statement 3 CONTINUE. The following is a description of the subroutine arguments:

XO x_p location of left-hand edge of main character in label,
in.

YO y_p location of centerline of main character in label, in.

HT height of main character, in.

ISYM identifying number of label

Subroutine LAM

The function of subroutine LAM is to draw the script "7" character used as the subscript for the rolling-moment coefficient label generated by subroutine COEFSY. This special character is not available as part of the standard character series contained in the CALCOMP plotting subprogram package.

Subroutine GRIDLN

The function of subroutine GRIDLN is to draw a series of horizontal and vertical grid lines within a specified frame size. This process is

accomplished using simple straight line plotting pen movements. The following is a description of the subroutine arguments:

XO,YO	x_p and y_p values for the lower left-hand corner of the frame, in.
XL	width of frame, in.
YH	height of frame, in.
NDIV	number of grid lines per inch

Subroutine RUNKEY

The function of subroutine RUNKEY is to draw a legend containing a list of run numbers for the data plotted and a corresponding list of symbols. The run numbers are spaced so that a corresponding list of descriptors can be typed on a sheet of gum-back paper using the IBM Executive model typewriter and then cut and attached to the legend adjacent to the list of run numbers. This particular typewriter is widely used at Langley to type figure titles and legends. After drawing the symbols and run numbers the words "Symbol" and "Run" are drawn and underlined above the corresponding lists. The user often desires to scale a series of plots to either a smaller or larger size using the magnification factors available as part of the postprocessor PLOT control card. Using these magnification factors would also cause a corresponding change in the legend which would prohibit the use of the IBM Executive Model typewriter to type the list of descriptors. To overcome this problem, a magnification factor, XM, has been included in this subroutine that adjusts the letter and number heights and spacings to compensate for the postprocessor scaling. The following subroutine arguments and parameters are used in this subroutine:

XO, YO	The x_p and y_p values of the lower left-hand corner of the legend, in.
--------	---

RUN	array containing input run numbers
LSYM	array containing input symbol order
ISIZE	symbol size
HT	letter and number character height, in.
NRNMAX	maximum allowable number of runs per sheet
XM	magnification factor for character and spacings

APPLICATION OF COMPUTER PROGRAM

The basic program variables have been dimensioned for a maximum of 10 runs per sheet, 10 plots per sheet, 50 data points per run, 5000 data points per test, and 300 wind-tunnel test parameters per data point. The following procedure outlines the changes necessary to either increase or decrease these basic program variables:

(1) To change the maximum number of allowable runs per sheet, change the dimension of the variable RUN and the value of the parameter NRMAX in the program PLOT.

(2) To change the maximum number of plots per sheet, change the dimension of the variables XPLT, YPLT, YTAPE, YOFFSET, YSCALE, YLABEL, XTAPE, XOFFSET, XSCALE, XLABEL, VAR, and NLOC and the value of the parameter NPLMAX in program PLOT.

(3) To change the maximum number of points per run, change the dimension of the variables VAR, ARRAY1, ARRAY2 and the value of the parameter NDMAX in program PLOT; change the dimension of the variables TIN, TEDT, TSAVE, YSAVE, IEDT, ITEMP, and ISAVE in subroutine EDIT; change the dimension of the variables TIN, ISAVE, TSAVE, IEDT, TEDT, IUN, and TUN in subroutine PRTEDT; change the dimension of the variables MX, MY, and A in subroutines CURPLT and DASHLN; and change the dimension of the variables XNEW, YNEW, COEF, and SDV and the value of the parameter NMAX in subroutine LSQPLT.

(4) To change the maximum number of allowable data points per test, change the dimension of the variable NPT and the value of the parameter NPMAX in program PLOT.

(5) To change the maximum number of allowable wind-tunnel test parameters per data point, change the dimension of the variable DATA and the value of the parameter IMAX in subroutine DATADK.

The best way to demonstrate the application of the program is to present and discuss the output for several sample cases. Two sample cases will be presented (1) a single plot with three variables per sheet and (2) four plots with a single variable per sheet. The input data for these two cases are presented in table III. Both cases contain one sheet namelist with identical sheet and run numbers. The tabulated output for the first case is presented in table IV and the plotted output, in figure 1. If the user desires to plot dashed lines instead of symbols with connecting solid lines, the call to subroutine CURPLT in program PLOT can be replaced with a call to subroutine DASHLN which will produce the plot presented in figure 2. In a like manner, if the user desires to least-squares curve fit the input data, the call to CURPLT can be replaced with a call to subroutine LSQPLT which will produce the plot presented in figure 3(a) for the least-squares polynomial option and in figure 3(b) for the least-squares cubic-spline option. The programming changes required to obtain these optional plots are very simple and require very little programming knowledge.

Some programming knowledge is required if the user desires to generate a report quality plot on blank (no grid) paper by drawing the appropriate grid lines, axis scale values and labels, and the legends. The input requirements for the subroutines that generate the various parts of the plot are relatively simple and, after a short period of time, the average user should become very proficient in their use. The computer code required to generate report

quality plots for the two sample cases presented in table III are presented in table VII and VIII, respectively, and the corresponding plots in figures 4 and 5. Each set of code was inserted in the PLOT program following card number 233 (see appendix A) and the original code from cards number 237 to 269 deleted. Careful examination of the code for both cases will reveal that the same general procedure was used to generate a particular plot. This procedure is outlined as follows:

- (1) position plotting pen and draw grid lines (call subroutine GRIDLN)
- (2) draw all zero lines (call subroutine ZEROLN)
- (3) draw scale values on x-axis (call subroutine AXISLB)
- (4) position and draw label for x-axis (call subroutine COEFSY)
- (5) repeat steps 3 and 4 for y-axis
- (6) position and draw legend (call subroutine RUNKEY)

The two types of plots presented represent only a small example of the wide variety of the types of plots that can be generated using the PLOT program as the baseline. Most engineers that test in the same facility generally prefer to present the experimental data taken during a typical test program in a well-established standard format. The computer code to generate these standard report quality plots can be written by a programmer with more experience and simply inserted in the PLOT program code by the user with either the EDIT, XEDIT, UPDATE, or MODIFY editing postprocessor commands.

CONCLUDING REMARKS

The wind-tunnel data plotting program PLOTWD described in this report has been used successfully at Langley for several years with very few reported execution problems. The required format for the input data is somewhat restrictive, but the required user input is very simple and the variety of plots generated numerous. With relatively minor code changes, the original version of the program can be changed to handle either larger or smaller

amounts of test data. The program code presented in this report was written especially for use on the Langley CDC computer system. The program can be adapted to other computer systems provided the system selected has a random-access file capability and a similar CALCOMP software package. A copy of the basic source code for this program can be obtained using the following control statements:

```
GET, PLOTWD/UN = 621288N
```

```
UPDATE,F,P = PLOTWD, S = SOURCE, I = 0, L = 0.
```

REFERENCES

1. Cline, A. K.: Scalar- and Planar-Valued Curve Fitting Using Splines Under Tension. Communications of the ACM, Vol. 17, No. 14, 1984, pp. 218-220.
2. Hildebrand, Frances B.: Methods of Applied Mathematics. Second Edition, Prentice-Hall, Inc., 1965.
3. Reinsch, Christian H.: Smoothing by Spline Functions. Numerische Mathematik, Vol. 10, No. 3, 1967, pp. 177-183.

APPENDIX A

COMPUTER LISTING OF WIND-TUNNEL DATA PLOTTING PROGRAM PLOTWD

This appendix contains a computer listing of the wind-tunnel data plotting program PLOTWD which consists of a main program, eighteen subroutines, and two function subprograms.

CARD NO.

1	PROGRAM PLOT(INPUT,OUTPUT,TAPE5=INPUT,TAPE1,TAPE2,TAPE7=1001,	PT	1
	1TAPE11=1001,TAPE12=1001)	PT	2
	C PROGRAM TO PLOT AND FAIR WIND TUNNEL DATA	PT	3
5	C CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	PT	4
	C *****	PT	5
	C *	PT	6
	C * -- DEFINITION OF SETUP DECK CARDS --	PT	7
10	C * CARD NO. DESCRIPTION FORMAT	PT	8
	C * 1 TEST - TEST NUMBER (F10.0)	PT	9
	C * 2 SHEETW - WIDTH OF PLOTTING SHEET (INCHES) (3F10.0)	PT	10
	C * SHEETH - HEIGHT OF PLOTTING SHEET (INCHES)	PT	11
	C * SPACE - SPACE BETWEEN PLOTTING SHEETS (INCHES)	PT	12
15	C * 3 ISYM - STARTING SYMBOL NUMBER (2F10.0)	PT	13
	C * ISIZE - SYMBOL SIZE (1 = SMALL 2 = MEDIUM 3 = LARGE)	PT	14
	C * 4 IOP - PLOTTING OPTION IOP=0 PLOT DATA ONLY (2F10.0)	PT	15
	C * IOP=1 PLOT AND FAIR DATA	PT	16
20	C * TENSION - SPLINE TENSION FACTOR (FOR IOP=1 ONLY)	PT	17
	C * 5 IEDIT - EDIT OPTION IEDIT=0 NO EDITING (2F10.0)	PT	18
	C * IEDIT=1 EDIT WITH GIVEN TOLERANCE	PT	19
25	C * TOLR - EDIT TOLERANCE FOR INDEPENDENT VARIABLE	PT	20
	C * 6 IW - DATA ARRAY LOCATION OF INDEPENDENT VARIABLE (F10.0)	PT	21
30	C * 7 NPLOT - NUMBER OF PLOTS PER SHEET (F10.0)	PT	22
	C * 8 YTAPE - ARRAY LOCATION OF Y VARIABLE (2(3F10.0,A10))	PT	23
35	C * YOFFSET - ORIGIN OFFSET OF Y VARIABLE (INCHES)	PT	24
	C * YSCALE - SCALE VALUE PER INCH FOR Y VARIABLE	PT	25
	C * YLABEL - LABEL FOR Y VARIABLE	PT	26
	C * XTAPE - ARRAY LOCATION OF X VARIABLE	PT	27
40	C * XOFFSET - ORIGIN OFFSET OF X VARIABLE (INCHES)	PT	28
		PT	29
		PT	30
		PT	31
		PT	32
		PT	33
		PT	34
		PT	35
		PT	36
		PT	37
		PT	38
		PT	39
		PT	40

CARD NO.

41	C *	XSCALE - SCALE VALUE PER INCH FOR X VARIABLE	* PT	41
	C *	XLABEL - LABEL FOR X VARIABLE	* PT	42
	C *	NOTE - CARD 8 IS REPEATED NPLOT TIMES	* PT	43
	C *		* PT	44
45	C *	-- DEFINITION OF SHEET NAMELIST PARAMETERS --	* PT	45
	C *		* PT	46
	C *	\$SHEET - NAMELIST LABEL	* PT	47
	C *	NO - SHEET NUMBER	* PT	48
	C *	RUN - RUN NUMBERS TO BE PLOTTED ON SHEET (MAXIMUM OF 10)	* PT	49
50	C *	NEWCASE - IF NEW SETUP DECK FOLLOWS SET NEWCASE=1	* PT	50
	C *	\$END - END OF NAMELIST	* PT	51
	C *		* PT	52
	C	*****	PT	53
	C		PT	54
55		INTEGER RUN	PT	55
	C		PT	56
	C	DIMENSION ARRAY LSYM WHICH CONTAINS ORDER OF PLOTTING SYMBOLS	PT	57
	C		PT	58
		DIMENSION LSYM(22)	PT	59
60	C		PT	60
	C	DIMENSION RUN(NRNMAX) WHERE NRNMAX IS THE MAXIMUM NUMBER OF	PT	61
	C	ALLOWABLE RUNS PER SHEET	PT	62
	C		PT	63
		DIMENSION RUN(10)	PT	64
65	C		PT	65
	C	DIMENSION NPT(NPMAX) WHERE NPMAX IS THE MAXIMUM NUMBER OF DATA	PT	66
	C	POINTS FOR TOTAL TEST	PT	67
	C	NOTE -- DO NOT DESTROY THIS ARRAY DURING PROGRAM EXECUTION	PT	68
	C		PT	69
70		DIMENSION NPT(5000)	PT	70
	C		PT	71
	C	DIMENSION EACH OF THE FOLLOWING ARRAYS BY NPLMAX WHICH IS THE	PT	72
	C	MAXIMUM NUMBER OF ALLOWABLE PLOTS PER SHEET	PT	73
	C		PT	74
75		DIMENSION XPLT(10), YPLT(10), YTAPE(10), YOFFSET(10), YSCALE(10),	PT	75
		1YLABFL(10), XTAPE(10), XOFFSET(10), XSCALE(10), XLABEL(10)	PT	76
	C		PT	77
	C	DIMENSION EACH OF THE FOLLOWING ARRAYS BY NDMAX WHICH IS THE	PT	78
	C	MAXIMUM NUMBER OF ALLOWABLE POINTS FOR A GIVEN RUN	PT	79
80	C		PT	80

CARD NO.

81		DIMENSION X(50), Y(50), T(50)	PT 81
	C		PT 82
	C	DIMENSION VAR(NDMAX,NVAR) AND NLOC(NVAR) WHERE NVAR=2*NPLMAX+1	PT 83
	C		PT 84
85		DIMENSION VAR(50,21), NLOC(21)	PT 85
	C		PT 86
	C	DIMENSION ARRAY1(7*NDMAX,GE.500) AND ARRAY2(3*NDMAX)	PT 87
	C		PT 88
		COMMON /WORK/ ARRAY1(500)	PT 89
90		COMMON /PLT/ ARRAY2(150)	PT 90
	C		PT 91
	C	SHEET NAMELIST PARAMETERS	PT 92
	C		PT 93
		NAMELIST /SHEET/ NO,RUN,NEWCASE	PT 94
95	C		PT 95
	C	DEFINE INPUT TAPE NUMBER	PT 96
	C		PT 97
		DATA JREAD/5/	PT 98
	C		PT 99
100	C	DEFINE ORDER OF PLOTTING SYMBOLS	PT 100
	C		PT 101
		DATA LSYM/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22	PT 102
		1/	PT 103
	C		PT 104
105	C	DEFINE MAXIMUM NUMBER OF ALLOWABLE RUNS PER SHEET	PT 105
	C		PT 106
		DATA NRNMAX/10/	PT 107
	C		PT 108
	C	DEFINE MAXIMUM NUMBER OF ALLOWABLE PLOTS PER SHEET	PT 109
110	C		PT 110
		DATA NPLMAX/10/	PT 111
	C		PT 112
	C	DEFINE MAXIMUM NUMBER OF ALLOWABLE DATA POINTS FOR A GIVEN RUN	PT 113
	C		PT 114
115		DATA NDMAX/50/	PT 115
	C		PT 116
	C	DEFINE MAXIMUM NUMBER OF ALLOWABLE TOTAL DATA POINTS	PT 117
	C		PT 118
		NPMAX=5000	PT 119
120	C		PT 120

CARD NO.

121	C	DEFINE PROGRAM OPTIONS	PT 121
	C		PT 122
	C	IF IPRT=0, PRINT EDITED DATA	PT 123
	C	IF IPRT=1, DO NOT PRINT EDITED DATA	PT 124
125		IPRT=0	PT 125
	C	LZERO=0, DRAW ZERO LINES	PT 126
	C	LZERO=1, DO NOT DRAW ZERO LINES	PT 127
		LZERO=1	PT 128
	C	NORG=0, DRAW ARROWHEAD AT ORIGIN OF PLOTTING VARIABLES	PT 129
130	C	NORG=1, DO NOT DRAW ARROWHEAD AT ORIGIN OF PLOTTING VARIABLES	PT 130
		NORG=0	PT 131
	C		PT 132
	C	SET PROGRAM CONSTANTS	PT 133
	C		PT 134
135		IDISK=0	PT 135
	C		PT 136
	C	INITIALIZE PLOTTING DEVICE	PT 137
	C		PT 138
		CALL PSEUDO	PT 139
140		CALL LEROY	PT 140
	C		PT 141
	C	READ SETUP DECK	PT 142
	C		PT 143
	1	CALL SETUP (ITEST,SHEETW,SHEETH,SPACE,ISYM,ISIZE,IOP,TENSION,IEDIT	PT 144
145		1,TCLR,IW,NPLOT,YTAPE,YOFFSET,YSCALE,YLABEL,XTAPE,XOFFSET,XSCALE,XL	PT 145
		2ABEL,JREAD,NPLMAX,IERR)	PT 146
		IF (IERR.NE.0) GO TO 23	PT 147
	C		PT 148
		REWIND 7	PT 149
150		REWIND 11	PT 150
		KEDT=0	PT 151
	C		PT 152
	C	LOAD DATA DISK	PT 153
	C		PT 154
155		IF (IDISK) 3,2,3	PT 155
	2	CALL DATADK (0,ITEST,IRUN,VAR,NVAR,NLOC,NP,NPT,NPMAX,NOMAX,IERR)	PT 156
		IF (IERR.NE.0) GO TO 23	PT 157
	C		PT 158
	3	IDISK=1	PT 159
160		PRINT 24	PT 160

CARD NO.

161	C		PT 161
	C	LOAD PLOTTING VARIABLES	PT 162
	C		PT 163
		DO 4 I=1,NPLOT	PT 164
165		J2=2*I	PT 165
		J1=J2-1	PT 166
		NLOC(J1)=IFIX(YTAPE(I)+0.0001)	PT 167
	4	NLOC(J2)=IFIX(XTAPE(I)+0.0001)	PT 168
		NVAR=2*NPLOT+1	PT 169
170		NLOC(NVAR)=IW	PT 170
	C		PT 171
	C	CHECK FOR DUPLICATE PLOTTING VARIABLES	PT 172
	C		PT 173
		DO 9 I=1,NPLOT	PT 174
175		YPLT(I)=XPLT(I)=0.	PT 175
		IF (I.EQ.1) GO TO 9	PT 176
		J1=I-1	PT 177
		DO 8 J=1,J1	PT 178
		IF (XTAPE(J)-XTAPE(I)) 6,5,6	PT 179
180	5	IF (XOFFSET(J).EQ.XOFFSET(I).AND.XSCALE(J).EQ.XSCALE(I)) XPLT(I)=1	PT 180
		1.0	PT 181
	6	IF (YTAPE(J)-YTAPE(I)) 8,7,8	PT 182
	7	IF (YOFFSET(J).EQ.YOFFSET(I).AND.YSCALE(J).EQ.YSCALE(I)) YPLT(I)=1	PT 183
		1.0	PT 184
185	8	CONTINUE	PT 185
	9	CONTINUE	PT 186
	C		PT 187
	C	READ SHEET NUMBER AND RUNS TO BE PLOTTED	PT 188
	C		PT 189
190	10	DO 11 I=1,NRNMAX	PT 190
	11	RUN(I)=0	PT 191
		NEWCASE=NO=0	PT 192
	C		PT 193
		READ (JREAD,SHEET)	PT 194
195		IF (EOF(JREAD)) 23,12	PT 195
	12	CONTINUE	PT 196
	C		PT 197
		PRINT 25, NO,RUN	PT 198
	C		PT 199
200	C	PERFORM GRID CHECK	PT 200

LISTING OF DECK: PLOT

PAGE 6

CARD NO.

201	C	CALL CALPLT (0.,0.,-3)	PT 201
		CALL GRIDCK	PT 202
	C		PT 203
205	C	POSITION PLOTTING PEN AT LOWER LEFT HAND CORNER OF PLOTTING SHEET	PT 204
	C		PT 205
		CALL CALPLT (2.,2.,-3)	PT 206
	C		PT 207
	C	LABEL TEST, SHEET, AND RUN NUMBERS	PT 208
210	C		PT 209
		SZ=0.15	PT 210
		D=-1.9	PT 211
	C	LABEL TEST NUMBFR	PT 212
		CALL NOTATE (D,-2.0,SZ,5HTEST,0.,5)	PT 213
215		D=30.*SZ/7.+D	PT 214
		TP=FLOAT(ITEST)	PT 215
		CALL NUMBER (D,-2.0,SZ,TP,0.,-1)	PT 216
		D=36.*SZ/7.+D	PT 217
	C	LABEL SHEET NUMBER	PT 218
220		CALL NOTATE (D,-2.0,SZ,7H,SHEET,0.,7)	PT 219
		D=42.*SZ/7.+D	PT 220
		TP=FLOAT(ND)	PT 221
		CALL NUMBER (D,-2.0,SZ,TP,0.,-1)	PT 222
		D=36.*SZ/7.+D	PT 223
225	C	LABEL RUN NUMBERS	PT 224
		CALL NOTATE (D,-2.0,SZ,6H,RUNS,0.,6)	PT 225
		D=42.*SZ/7.+D	PT 226
		DO 13 I=1,NRNMAX	PT 227
		TP=FLOAT(RUN(I))	PT 228
230		CALL NUMBER (D,-2.0,SZ,TP,0.,-1)	PT 229
	13	D=42.*SZ/7.+D	PT 230
		IX=IFIX(D)	PT 231
		DMAX=FLOAT(IX)+SPACE+2.0	PT 232
	C		PT 233
235	C	DRAW PLOTTING SHEET FRAME	PT 234
	C		PT 235
		CALL CALPLT (0.,0.,3)	PT 236
		CALL CALPLT (SHEETW,0.,2)	PT 237
		CALL CALPLT (SHEETW,SHEETH,2)	PT 238
240		CALL CALPLT (0.,SHEETH,2)	PT 239
			PT 240

CARD NO.

241		CALL CALPLT (0.,0.,2)	PT 241
	C		PT 242
	C	LABEL ORIGIN, VARIABLE DESCRIPTION, AND SCALE	PT 243
	C		PT 244
245		DO 15 I=1,NPLOT	PT 245
		IF (YPLT(I).EQ.1.) GO TO 14	PT 246
		D=YOFFSET(I)	PT 247
		IF (NORG.EQ.0) CALL PARROW (-0.3,D,0.,D,-2.0,.3)	PT 248
		CALL NOTATE (-1.8,D,0.15,YLABEL(I),0.,10)	PT 249
250		D=YOFFSET(I)-0.2	PT 250
		CALL NUMBER (-1.8,D,0.15,YSCALE(I),0.,4)	PT 251
	14	IF (XPLT(I).EQ.1.) GO TO 15	PT 252
		D=XOFFSET(I)	PT 253
		IF (NORG.EQ.0) CALL PARROW (D,-0.3,D,0.,-2.0,.3)	PT 254
255		D=XOFFSET(I)-0.65	PT 255
		CALL NOTATE (D,-0.6,0.15,XLABEL(I),0.,10)	PT 256
		CALL NUMBER (D,-0.8,0.15,XSCALE(I),0.,4)	PT 257
	15	CONTINUE	PT 258
	C		PT 259
260	C	DRAW ZERO LINES	PT 260
	C		PT 261
		IF (LZERO.NE.0) GO TO 18	PT 262
		DO 17 I=1,NPLOT	PT 263
		IF (YPLT(I).EQ.1.) GO TO 16	PT 264
265		CALL ZEROLN (0.,YOFFSET(I),SHEETW,1)	PT 265
	16	IF (XPLT(I).EQ.1.) GO TO 17	PT 266
		CALL ZEROLN (XOFFSET(I),0.,SHEETH,2)	PT 267
	17	CONTINUE	PT 268
	18	CONTINUE	PT 269
270	C		PT 270
	C	READ DATA FROM DATA DISK AND PLOT	PT 271
	C		PT 272
		KSYM=ISYM-1	PT 273
	C		PT 274
275		DO 21 IR=1,NRNMAX	PT 275
	C	DEFINE RUN NUMBER	PT 276
		IRUN=RUN(IR)	PT 277
	C	DEFINE PLOTTING SYMBOL	PT 278
		KSYM=KSYM+1	PT 279
280		IF (KSYM.GT.22) KSYM=ISYM	PT 280

CARD NO.

281		IF (KSYM.LT.1) KSYM=1	PT 281
		NSYM=LSYM(KSYM)	PT 282
	C		PT 283
		IF (IRUN.EQ.0) GO TO 21	PT 284
285	C	READ DATA FROM DATA DISK	PT 285
		CALL DATADK (1,ITEST,IRUN,VAR,NVAR,NLOC,NP,NPT,NPMAX,NDMAX,IERR)	PT 286
		IF (IERR.NE.0) GO TO 21	PT 287
	C	LOAD AND PLOT DATA	PT 288
		DO 20 I=1,NPLOT	PT 289
290		J2=2*I	PT 290
		J1=J2-1	PT 291
		IX=NLOC(J2)	PT 292
	C	LOAD PLOTTING ARRAYS	PT 293
		N=0	PT 294
295		DO 19 J=1,NP	PT 295
		XP=VAR(J,J2)/XSCALE(I)+XOFFSET(I)	PT 296
		YP=VAR(J,J1)/YSCALE(I)+YOFFSET(I)	PT 297
		TP=VAR(J,NVAR)	PT 298
	C	CHECK TO SEE IF DATA IS WITHIN PLOTTING FRAME	PT 299
300		IF (XP.LT.0.0.OR.XP.GT.SHEETW) GO TO 19	PT 300
		IF (YP.LT.0.0.OR.YP.GT.SHEETH) GO TO 19	PT 301
		N=N+1	PT 302
		X(N)=XP	PT 303
		Y(N)=YP	PT 304
305		T(N)=TP	PT 305
	19	CONTINUE	PT 306
	C	PRINT MESSAGE IF SOME POINTS NOT WITHIN PLOTTING FRAME	PT 307
		JS=NP-N	PT 308
		IF (N.NE.NP) PRINT 26, IRUN,JS,I	PT 309
310	C	EDIT DATA	PT 310
		IERR=0	PT 311
		IF (IEDIT.NE.0) CALL EDIT (T,X,Y,N,NSYM,ISIZE,IRUN,TOLP,KEDT,IERR)	PT 312
		IF (IERR.NE.0) GO TO 20	PT 313
	C	PLOT VARIABLES	PT 314
315		JS=0	PT 315
		IF (IX.NE.IW) JS=1	PT 316
		CALL CURPLT (T,X,Y,N,JS,NSYM,ISIZE,IOP,IRUN,TENSION)	PT 317
	C	DO NEXT VARIABLE	PT 318
	20	CONTINUE	PT 319
320	C	DO NEXT RUN	PT 320

CARD NO.

321	21	CONTINUE	PT 321
	C	ADVANCE TO ORIGIN OF NEXT FRAME	PT 322
		D=SHEFTW+SPACE+2.0	PT 323
		IF (D.LT.DMAX) D=DMAX	PT 324
325		CALL NFRAME (D,0.)	PT 325
	C		PT 326
	C	READ NEXT SHEET NAMELIST UNLESS NEW SETUP DECK FOLLOWS	PT 327
	C		PT 328
		IF (NEWCASE.NE.0) GO TO 22	PT 329
330		GO TO 10	PT 330
	C		PT 331
	C	PRINT EDITED DATA	PT 332
	C		PT 333
	22	IF (IEDIT.NE.0.AND.IPRT.EQ.0) CALL PRTEDT	PT 334
335		PRINT 27	PT 335
		GO TO 1	PT 336
	23	IF (IEDIT.NE.0.AND.IPRT.EQ.0) CALL PRTEDT	PT 337
	C		PT 338
	C	FINALIZE PLOTTING DEVICE	PT 339
340	C		PT 340
		CALL NOTATE (2.,2.,0.5,15HEND OF PLOTTING,90.,15)	PT 341
		CALL NFRAME (4.,0.)	PT 342
		CALL CALPLT (0.,0.,999)	PT 343
	C		PT 344
345	C	PRINT CLOSING MESSAGE	PT 345
	C		PT 346
		PRINT 28	PT 347
		STOP	PT 348
	C		PT 349
350	24	FORMAT (1H1,3X,12HDATA PLOTTED)	PT 350
	25	FORMAT (/4X,10HSHEET NO =,I5,5X,6HRUNS =,9(I7,1H,),I7/(30X,9(I7,1H	PT 351
		1,),I7))	PT 352
	26	FORMAT (/4X,9HFOR RUN =,I7,3H , ,I4,55H POINTS WERE NOT WITHIN PLO	PT 353
		TTING SHEET FRAME FOR PLOT =,I3)	PT 354
355	27	FORMAT (/4X,22HNEW CASE SHOULD FOLLOW)	PT 355
	28	FORMAT (/4X,22HALL PLOTTING COMPLETED)	PT 356
		END	PT 357-

CARD NO.

1		SUBROUTINE SETUP (ITEST,SHEETW,SHEETH,SPACE,ISYM,TSIZE,IOP,TENSION	ST	1
		1,IEDIT,TOLR,IW,NPLOT,YTAPE,YOFFSET,YSCALE,YLABEL,XTAPE,XOFFSET,XSC	ST	2
		2ALE,XLABEL,JREAD,NPLMAX,IERR)	ST	3
	C		ST	4
5	C	ROUTINE TO READ PLOTTING SETUP DECK	ST	5
	C		ST	6
	C	CODED BY -- HARRY L. MORGAN	ST	7
	C	NASA/LARC/TAD/AAB 1983	ST	8
	C		ST	9
	C	PARAMETER DEFINITION	ST	10
10	C	ITEST - TEST NUMBER	ST	11
	C	SHEETW - WIDTH OF PLOTTING SHEET (IN)	ST	12
	C	SHEETH - HEIGHT OF PLOTTING SHEET (IN)	ST	13
	C	SPACE - SPACE BETWEEN PLOTTING SHEETS (IN)	ST	14
	C	ISYM - STARTING SYMBOL NUMBER	ST	15
15	C	IOP - PLOTTING OPTION - IOP=0, PLOT SYMBOLS ONLY	ST	16
	C	IOP=1, PLOT AND FAIR DATA	ST	17
	C	TENSION - SPLINE TENSION FACTOR	ST	18
	C	IEDIT - EDIT OPTION - IEDIT=0, DO NOT SORT AND EDIT DATA	ST	19
	C	IEDIT=1, SORT AND EDIT DATA	ST	20
20	C	TOLR - EDIT TOLERANCE OF INDEPENDENT VARIABLE	ST	21
	C	IW - DATA ARRAY LOCATION OF INDEPENDENT VARIABLE	ST	22
	C	NPLOT - NUMBER OF PLOTS PER SHEET	ST	23
	C	YTAPE - DATA ARRAY LOCATION OF Y VARIABLE	ST	24
	C	YOFFSET - ORIGIN OF Y VARIABLE (IN)	ST	25
25	C	YSCALE - VALUE PER INCH FOR Y VARIABLE	ST	26
	C	YLABEL - LABEL FOR Y VARIABLE	ST	27
	C	XTAPE - DATA ARRAY LOCATION OF X VARIABLE	ST	28
	C	XOFFSET - ORIGIN OF X VARIABLE (IN)	ST	29
	C	XSCALE - VALUE PER INCH FOR X VARIABLE	ST	30
30	C	XLABEL - LABEL FOR X VARIABLE	ST	31
	C	JREAD - TAPE NUMBER OF INPUT FILE	ST	32
	C	IERR - ERROR INDICATOR - IERR=0, NO ERRORS OCCURRED	ST	33
	C	IERR=1, ERRORS OCCURRED	ST	34
35	C	XTAPE,XOFFSET,XSCALE,XLABEL,YTAPE,YOFFSET,YSCALE, AND YLABEL	ST	35
	C	MUST BE PROPERLY DIMENSIONED IN CALLING PROGRAM	ST	36
	C	DIMENSION XTAPE(1), XOFFSET(1), XSCALE(1), XLABEL(1), YTAPE(1), YO	ST	37
	C	FFSET(1), YSCALE(1), YLABEL(1)	ST	38
	C	DIMENSION AND DEFINE SYMBOL SIZE ARRAY AND NMAX	ST	39
40	C	DIMENSION IS(3)	ST	40

CARD NO.

41		DATA IS/6HSMALL,6HMEDIUM,6HLARGE /	ST	41
	C		ST	42
	C	INITIALIZE ERROR INDICATOR	ST	43
	C		ST	44
45		IERP=0	ST	45
	C		ST	46
	C	READ AND PRINT TEST NUMBER	ST	47
	C		ST	48
		READ (JREAD,6) D1	ST	49
50		IF (EOF(JREAD)) 4,1	ST	50
	1	ITEST=IFIX(D1+0.0001)	ST	51
		PRINT 7, ITEST	ST	52
	C		ST	53
	C	READ AND PRINT SHEETW, SHEETH, AND SPACE	ST	54
55	C		ST	55
		READ (JREAD,6) SHEETW,SHEETH,SPACE	ST	56
		IF (SHEETW.LT.11.0) SHEETW=11.0	ST	57
		PRINT 8, SHEETW,SHEETH,SPACE	ST	58
	C		ST	59
60	C	READ AND PRINT ISYM AND ISIZE	ST	60
	C		ST	61
		READ (JREAD,6) D1,D2	ST	62
		ISYM=IFIX(D1+0.0001)	ST	63
		ISIZE=IFIX(D2+0.0001)	ST	64
65		IF (ISYM.LT.1.OR.ISYM.GT.22) ISYM=1	ST	65
		IF (ISIZE.LT.1.OR.ISIZE.GT.3) ISIZE=2	ST	66
		PRINT 9, ISYM,IS(ISIZE)	ST	67
	C		ST	68
	C	READ AND PRINT IOP AND TENSION	ST	69
70	C		ST	70
		READ (JREAD,6) D1,TENSION	ST	71
		IOP=IFIX(D1+0.0001)	ST	72
		IF (IOP.LT.0.OR.IOP.GT.1) IOP=0	ST	73
		IF (IOP.EQ.0) PRINT 10	ST	74
75		IF (IOP.EQ.1) PRINT 11, TENSION	ST	75
	C		ST	76
	C	READ AND PRINT IEDIT AND TOLR	ST	77
	C		ST	78
		READ (JREAD,6) D1,TOLR	ST	79
80		IEDIT=IFIX(D1+0.0001)	ST	80

CARD NO.

81		IF (IEDIT.LT.0.OR.IEDIT.GT.1) IFDIT=0	ST 81
		IF (IEDIT.EQ.1) PRINT 12, TOLR	ST 82
	C		ST 83
	C	READ AND PRINT IW	ST 84
85	C		ST 85
		READ (JREAD,6) D1	ST 86
		IW=IFIX(D1+0.0001)	ST 87
		PRINT 13,IW	ST 88
		IF (IW.LE.0) GO TO 4	ST 89
90	C		ST 90
	C	READ AND PRINT NPL0T	ST 91
	C		ST 92
		READ (JREAD,6) D1	ST 93
		NPL0T=IFIX(D1+0.0001)	ST 94
95		PRINT 14, NPL0T	ST 95
		IF (NPL0T.LE.0.OR.NPL0T.GT.NPLMAX) GO TO 3	ST 96
	C		ST 97
	C	READ AND PRINT YTAPE,YOFFSET,YSCALE,YLABEL,XTAPE,XOFFSET,XSCALE,	ST 98
	C	AND XLABEL	ST 99
100	C		ST 100
		DO 2 I=1,NPL0T	ST 101
		READ (JREAD,15) YTAPE(I),YOFFSET(I),YSCALE(I),YLABEL(I),XTAPE(I),X	ST 102
		10FFSET(I),XSCALE(I),XLABEL(I)	ST 103
		IX=IFIX(XTAPE(I)+0.0001)	ST 104
105		IY=IFIX(YTAPE(I)+0.0001)	ST 105
		PRINT 16, I,IY,YOFFSET(I),YSCALE(I),YLABEL(I),IX,XOFFSET(I),XSCALE	ST 106
		1(I),XLABEL(I)	ST 107
	2	CONTINUE	ST 108
		GO TO 5	ST 109
110	C		ST 110
	C	PRINT ERROR MESSAGES THAT MAY HAVE OCCURRED	ST 111
	C		ST 112
	3	PRINT 17, NPL0T	ST 113
		IERP=1	ST 114
115		GO TO 5	ST 115
	4	IERP=1	ST 116
	C		ST 117
	C	RETURN TO CALLING PROGRAM	ST 118
	C		ST 119
120	5	RETURN	ST 120

LISTING OF DECK: SETUP

CARD NO.

121	C		ST 121
	6	FORMAT (4F10.4)	ST 122
	7	FORMAT (1H1,3X,30HPLOTTING SETUP DECK FOR TEST =,I5)	ST 123
	8	FORMAT (4X,13HSHEET WIDTH =,F8.2,5X,14HSHEET HEIGHT =,F8.2,5X,22HS	ST 124
	9	1SPACE BETWEEN SHEETS =,F8.2)	ST 125
125		FORMAT (4X,24HSTARTING SYMBOL NUMBER =,I3,5X,15HSYMBOL SIZE IS ,A6	ST 126
	10	1)	ST 127
	11	FORMAT (4X,25HDATA WILL BE PLOTTED ONLY)	ST 128
	12	FORMAT (4X,59HDATA WILL BE FAIRED AND PLOTTED USING SPLINE WITH TE	ST 129
130		1NSION =,F6.2)	ST 130
	13	FORMAT (4X,44HDATA WILL BE EDITED WITH AN EDIT TOLERANCE =,FR.3)	ST 131
	14	FORMAT (4X,39HTAPE LOCATION OF INDEPENDENT VARIABLE =,I4)	ST 132
	15	FORMAT (4X,27HNUMBER OF PLOTS PER SHEET =,I4/7X,8HPLOT NO.,5X,5HYT	ST 133
	16	1APE,5X,7HYOFFSET,6X,6HYSCALE,6X,6HYLABEL,9X,5HXTAPE,5X,7HXOFFSET,6	ST 134
	17	2X,6HXSCALE,6X,6HXLABEL)	ST 135
135		FORMAT (3F10.4,A10,3F10.4,A10)	ST 136
	15	FORMAT (8X,I3,1X,2(8X,I4,5X,F7.2,2X,F12.4,2X,A10))	ST 137
	16	FORMAT (/4X,40HERROR OCCURRED -- IN SETUP DECK NPLNT =,I4)	ST 138
	17	END	ST 139-

CARD NO.

1		SUBROUTINE DATADK (ICODE, ITEST, IPUN, VAR, NVAR, NLOC, NP, NPT, NPMAX, NDM	DK	1
		1AX, IERR)	DK	2
	C		DK	3
	C	ROUTINE TO READ FROM OR WRITE TO A DATA DISK	DK	4
5	C		DK	5
	C	CODED BY -- HARRY L. MORGAN	DK	6
	C	NASA/LARC/TAD/AAB	DK	7
	C	1983	DK	8
	C	PARAMETER DEFINITION	DK	9
	C		DK	10
10	C	ICODE - READ OR WRITE OPTION	DK	11
	C	ITEST - TEST NUMBER	DK	12
	C	ICODE=0, WRITE IN DATA DISK	DK	13
	C	ICODE=1, READ DATA FROM DATA DISK	DK	14
	C	IRUN - RUN NUMBER TO BE READ FROM DATA DISK	DK	15
15	C	VAR(I,J) - TWO DIMENSIONAL ARRAY CONTAINING DATA READ FROM	DK	16
	C	DATA DISK. (I=1,.....,NP AND J=1,.....,NVAR)	DK	17
	C	NVAR - NUMBER OF VARIABLES READ FROM DISK FOR EACH POINT OF	DK	18
	C	DATA	DK	19
	C	NLOC(J) - ARRAY CONTAINING DATA ARRAY LOCATIONS OF DESIRED	DK	20
20	C	VARIABLES. (J=1,.....,NVAR)	DK	21
	C	NP - NUMBER OF DATA POINTS READ FROM DISK FOR DESIRED RUN	DK	22
	C	NPT - ARRAY CONTAINING COUNT NUMBER OF DATA POINTS STORED ON	DK	23
	C	THE DATA DISK	DK	24
	C	NPMAX - MAXIMUM NUMBER OF DATA POINTS	DK	25
25	C	NDMAX - MAXIMUM NUMBER OF ALLOWABLE DATA POINTS FOR ANY GIVEN	DK	26
	C	RUN	DK	27
	C	IERR - ERROR INDICATOR	DK	28
	C	IERR=0, NO ERRORS OCCURRED	DK	29
	C	IERR=1, ERROR OCCURRED DURING READ OR WRITE	DK	30
30	C	IERR=2, IRUN NOT ON DATA DISK	DK	31
	C		DK	32
	C	NOTE -- VAR, NLOC, AND NPT MUST BE PROPERLY DIMENSIONED IN THE	DK	33
	C	CALLING PROGRAM	DK	34
35	C	DIMENSION VAR(NDMAX,1), NLOC(1), NPT(1)	DK	35
	C		DK	36
	C	WORK ARRAY	DK	37
	C		DK	38
	C	COMMON /PLT/ NPRT(10)	DK	39
40	C		DK	40

CARD NO.

41	C	DIMENSION DATA(IMAX+3) WHERE IMAX IS THE MAXIMUM NUMBER OF DATA	DK	41
	C	VARIABLES STORED ON THE INPUT DATA TAPE	DK	42
	C		DK	43
		COMMON /WORK/ DATA(303)	DK	44
45	C		DK	45
	C	DEFINE ITAPE (INPUT DATA TAPE NUMBER) AND NDISK (DATA DISK	DK	46
	C	NUMBER)	DK	47
	C		DK	48
		DATA ITAPE/1/,NDISK/2/	DK	49
50	C	DEFINE IMAX	DK	50
		DATA IMAX/300/	DK	51
	C		DK	52
	C	COMPUTE ROUTINE CONSTANTS	DK	53
	C		DK	54
55		IERR=0	DK	55
		I3=IMAX+3	DK	56
		REWIND 12	DK	57
	C		DK	58
	C	SELECT READ OR WRITE PROCEDURE	DK	59
60	C		DK	60
		IF (ICODE) 1,1,16	DK	61
	C		DK	62
	C	WRITE DATA ON DATA DISK	DK	63
	C		DK	64
65	C	INITIALIZE NPT	DK	65
	1	DO 2 I=1,NPMAX	DK	66
	2	NPT(I)=0	DK	67
	C	OPEN DATA DISK	DK	68
		CALL OPENMS (NDISK,NPT,NPMAX,0)	DK	69
70	C	COMPUTE NEEDED CONSTANTS	DK	70
		I1=IMAX+1	DK	71
		K=0	DK	72
		IRUNP=0	DK	73
		IC=0	DK	74
75	C	REWIND INPUT TAPE	DK	75
		REWIND ITAPE	DK	76
	C	READ DATA FROM INPUT TAPE	DK	77
	3	READ (ITAPE) IRUN,ITESTC,IPT,(DATA(I),I=1,IMAX)	DK	78
		IF (EOF(ITAPE)) 10,4	DK	79
80	4	CONTINUE	DK	80

LISTING OF DECK: DATADK

PAGE 3

CARD NO.

81		IF (IRUN.GT.99999) GO TO 10	DK 81
		IF (ITESTC.NE.ITEST) GO TO 15	DK 82
	C		DK 83
	C	NOTE -- POINTS OR RUNS CAN BE VOIDED HERE	DK 84
85	C		DK 85
	C	LOAD DATA ARRAY	DK 86
		DO 5 I=1,IMAX	DK 87
		J=I1-I	DK 88
		J3=J+3	DK 89
90		DATA (J3)=DATA(J)	DK 90
	5	CONTINUE	DK 91
		DATA (1)=FLOAT(IRUN)	DK 92
		DATA (2)=FLOAT(ITESTC)	DK 93
		DATA (3)=FLOAT(IPT)	DK 94
95		K=K+1	DK 95
	C	CHECK TO SEE IF POINT COUNT EXCEEDS ALLOWABLE MAXIMUM	DK 96
		IF (K-NPMAX) 6,14,14	DK 97
	C	WRITE ON TAPE 12 RUN NUMBER AND COUNT NUMBER OF FIRST DATA	DK 98
	C	POINT OF RUN	DK 99
100	6	IF (IRUNP) 7,8,7	DK 100
	7	IF (IRUN-IRUNP) 8,9,8	DK 101
	8	IC=IC+1	DK 102
		WRITE (12) IRUN,K	DK 103
		IRUNP=IRUN	DK 104
105	C	WRITE DATA ON DATA DISK	DK 105
	9	CALL WRITMS (NDISK,DATA,I3,K,0,0)	DK 106
	C	READ NEXT DATA POINT ON INPUT TAPE	DK 107
		GO TO 3	DK 108
	C	WRITE ON TAPE 12 RUN NUMBER AND COUNT NUMBER OF LAST DATA	DK 109
110	C	POINT + 1 OF RUN	DK 110
	10	K=K+1	DK 111
		WRITE (12) IRUNP,K	DK 112
		END FILE 12	DK 113
	C		DK 114
115	C	PRINT SUMMARY OF RUNS STORED ON DATA DISK	DK 115
	C		DK 116
		PRINT 29, ITESTC	DK 117
		IF (IC.EQ.0) GO TO 13	DK 118
		REWIND 12	DK 119
120		K=0	DK 120

CARD NO.

121		DO 12 I=1,IC	DK 121
		READ (12) IRUN,JS	DK 122
		K=K+1	DK 123
		NPRT(K)=IRUN	DK 124
125		IF (K-10) 12,11,11	DK 125
	11	PRINT 30, (NPRT(J),J=1,10)	DK 126
		K=0	DK 127
	12	CONTINUE	DK 128
		IF (K.GT.0.AND.K.LT.10) PRINT 30, (NPRT(J),J=1,K)	DK 129
130		READ (12) IRUN,K	DK 130
		K=K-1	DK 131
		PRINT 31, K	DK 132
		REWIND 12	DK 133
		GO TO 28	DK 134
135	C		DK 135
	C	PRINT ERROR THAT MAY HAVE OCCURRED DURING WRITE TO DISK	DK 136
	C		DK 137
	13	PRINT 32	DK 138
		IERR=1	DK 139
140		GO TO 28	DK 140
	14	PRINT 33, NPMAX	DK 141
		IERR=1	DK 142
		GO TO 28	DK 143
	15	PRINT 34, ITEST,ITESTC	DK 144
145		IERR=1	DK 145
		GO TO 28	DK 146
	C		DK 147
	C	READ DATA FROM DATA DISK	DK 148
	C		DK 149
150	C	READ TAPE 12 TO FIND POINT COUNT OF FIRST DATA POINT FOR	DK 150
	C	DESIRED RUN	DK 151
	16	READ (12) J,K	DK 152
		IF (EOF(12)) 23,17	DK 153
	17	IF (J-IRUN) 16,18,16	DK 154
155	C	READ TAPE 12 TO FIND POINT COUNT OF FIRST DATA POINT FOR	DK 155
	C	NEXT RUN ON DATA DISK	DK 156
	18	READ (12) J,JS	DK 157
		IF (EOF(12)) 24,19	DK 158
	C	COMPUTE NUMBER OF DATA POINTS ON DATA DISK FOR DESIRED RUN	DK 159
160	19	NP=JS-K	DK 160

CARD NO.

161		IF (NP.GT.NDMAX) GO TO 26	DK 161
	C	READ DATA FROM DATA DISK	DK 162
		DO 22 I=1,NP	DK 163
		CALL READMS (NDISK,DATA,I3,K)	DK 164
165	C	CHECK TEST NUMBER	DK 165
		ITESTC=IFIX(DATA(2)+0.0001)	DK 166
		IF (ITESTC.NE.ITEST) GO TO 25	DK 167
	C	STORE DATA NEEDED FOR PLOTTING	DK 168
		DO 21 J=1,NVAR	DK 169
170		J3=NLOC(J)+3	DK 170
		IF (J3-I3) 20,20,27	DK 171
	20	VAR(I,J)=DATA(J3)	DK 172
	21	CONTINUE	DK 173
	22	K=K+1	DK 174
175		GO TO 28	DK 175
	C		DK 176
	C	PRINT ERROR THAT MAY HAVE OCCURRED DURING READ FROM DATA DISK	DK 177
	C		DK 178
	23	PRINT 35, IRUN	DK 179
180		IERR=2	DK 180
		GO TO 28	DK 181
	24	PRINT 36, IRUN	DK 182
		IERR=1	DK 183
		GO TO 28	DK 184
185	25	PRINT 37, ITEST,ITESTC	DK 185
		IERR=1	DK 186
		GO TO 28	DK 187
	26	PRINT 38, IRUN,NDMAX	DK 188
		IERR=1	DK 189
190		GO TO 28	DK 190
	27	PRINT 39, JS,IRUN,IMAX	DK 191
		IERR=1	DK 192
	C		DK 193
	C	RETURN TO CALLING PROGRAM	DK 194
195	C		DK 195
	28	RETURN	DK 196
	C		DK 197
	29	FORMAT (1H1,5X,71HTHE FOLLOWING LIST OF RUNS ARE AVAILABLE ON THE	DK 198
		1DATA DISK FOR TEST NO. ,I5/)	DK 199
200	30	FORMAT (5X,10T10)	DK 200

CARD NO.

201	31	FORMAT (/5X,29HTOTAL NUMBER OF DATA POINTS =,I7)	DK 201
	32	FORMAT (/5X,67HERROR OCCURRED DURING WRITE TO DATA DISK --- NO DA ITA ON INPUT TAPE)	DK 202
	33	FORMAT (/5X,92HERROR OCCURRED DURING WRITE TO DATA DISK --- POINT	DK 203
205		1 COUNT HAS EXCEEDED ALLOWABLE MAXIMUM OF ,I7)	DK 204
	34	FORMAT (/5X,71HERROR OCCURRED DURING WRITE TO DATA DISK --- REQUE	DK 205
		1STED TEST NUMBER IS ,I5,34H AND TEST NUMBER ON INPUT TAPE IS ,I5)	DK 206
	35	FORMAT (/5X,58HERROR OCCURRED DURING READ FROM DATA DISK --- RUN	DK 207
		1NUMBER ,I7,15H IS NOT ON DISK)	DK 208
210	36	FORMAT (/5X,91HERROR OCCURRED DURING READ FROM DATA DISK --- SEAR	DK 209
		1CH FOR STARTING POINT FOR RUN FOLLOWING ,I7,14H WAS NOT FOUND)	DK 210
	37	FORMAT (/5X,72HERROR OCCURRED DURING READ FROM DATA DISK --- REQU	DK 211
		1ESTED TEST NUMBER IS ,I5,28H AND TEST NUMBER ON DISK IS ,I5)	DK 212
	38	FORMAT (/5X,47HERROR OCCURRED DURING READ FROM DATA DISK --- /5X,	DK 213
215		125HNUMBER OF POINTS FOR RUN ,I7,30H EXCEEDS ALLOWABLE MAXIMUM OF ,	DK 214
		2I4)	DK 215
	39	FORMAT (/5X,47HERROR OCCURRED DURING READ FROM DATA DISK --- /5X,	DK 216
		118Hvariable LOCATION ,I4,23H REQUESTED FOR RUN NO. ,I7,30H EXCEEDS	DK 217
		2 ALLOWABLE MAXIMUM OF ,I4)	DK 218
220		END	DK 219
			DK 220-

LISTING OF DECK: EDIT

PAGE 1

CARD NO.

1		SUBROUTINE EDIT (T,X,Y,N,NSYM,ISIZE,IRUN,TOLR,KEDT,IFERR)	ED	1
	C		ED	2
	C		ED	3
	C	ROUTINE TO SORT AND EDIT DATA	ED	4
5	C		ED	5
	C	CODED BY -- HARRY L. MORGAN	ED	6
	C	NASA/LARC/TAD/AA9	ED	7
	C	1983	ED	8
	C	PARAMETER DEFINITION	ED	9
	C	T - ARRAY CONTAINING INDEPENDENT VARIABLE	ED	10
10	C	X - ARRAY CONTAINING X VARIABLE	ED	11
	C	Y - ARRAY CONTAINING Y VARIABLE	ED	12
	C	N - NUMBER OF POINTS TO BE SORTED AND EDITED	ED	13
	C	NSYM - SYMBOL NUMBER TO BE USED TO PLOT UNEDITED POINTS	ED	14
	C	ISIZE - SYMBOL SIZE	ED	15
15	C	IRUN - RUN NUMBER	ED	16
	C	TOLR - EDIT TOLERANCE OF INDEPENDENT VARIABLE	ED	17
	C	KEDT - NUMBER OF RUNS ALREADY EDITED AND STORED FOR FUTURE	ED	18
	C	PRINT OUT	ED	19
	C	IFERR - EDIT ERROR INDICATOR	ED	20
20	C	IFERR=0, NO ERRORS OCCURRED	ED	21
	C	IFERR=1, DURING EDIT OF FIRST SORT POINT ALL GREATER	ED	22
	C	POINTS ARE WITHIN EDIT TOLERANCE	ED	23
	C	IFERR=2, DURING EDIT OF LAST SORT POINT ALL LESSER	ED	24
	C	POINTS ARE WITHIN EDIT TOLERANCE	ED	25
25	C	IFERR=3, DURING EDIT OF INTERMEDIATE SORT POINT ALL	ED	26
	C	LESSER POINTS ARE WITHIN EDIT TOLERANCE	ED	27
	C	IFERR=4, DURING EDIT OF INTERMEDIATE SORT POINT ALL	ED	28
	C	GREATER POINTS ARE WITHIN EDIT TOLERANCE	ED	29
	C		ED	30
30	C	T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING PROGRAM	ED	31
	C	DIMENSION T(1), X(1), Y(1)	ED	32
	C	WORK ARRAYS	ED	33
	C	COMMON /WORK/ TIN(50),TEDT(50),TSAVE(50),YSAVE(50),IEDT(50),ITEMP(ED	34
	C	150),ISAVE(50)	ED	35
35	C		ED	36
	C	CHECK TO SEE IF THIS RUN HAS ALREADY BEEN EDITED AND STORED FOR	ED	37
	C	FUTURE PRINT OUT	ED	38
	C		ED	39
	C	IFERR=0	ED	40
40	C	IF (N.LT.1) RETURN		

CARD NO.

41		NSTORE=1	ED	41
		REWIND 11	ED	42
		IF (KEDT) 3,3,1	ED	43
	1	READ (11) NRUN	ED	44
45		IF (EOF(11)) 3,2	ED	45
	2	IF (IRUN-NRUN) 1,4,1	ED	46
	3	KEDT=KEDT+1	ED	47
		NSTORE=0	ED	48
		IF (KEDT.GT.1) BACKSPACE 11	ED	49
50		WRITE (11) IRUN	ED	50
		END FILE 11	ED	51
	4	IN=N	ED	52
	C		ED	53
	C	SET INITIAL VALUES OF TIN, TEDT, IEDT, AND ITEMP	ED	54
55	C		ED	55
		DO 5 I=1,N	ED	56
		TIN(I)=T(I)	ED	57
		TEDT(I)=0.0	ED	58
		IEDT(I)=0	ED	59
60	5	ITEMP(I)=I	ED	60
	C		ED	61
	C	SORT DATA	ED	62
	C		ED	63
		K1=N	ED	64
65		DO 12 I=1,N	ED	65
		IF (I.EQ.1) GO TO 7	ED	66
		K=0	ED	67
		DO 6 JJ=1,K1	ED	68
		IF (JJ.EQ.JSAVE) GO TO 6	ED	69
70		K=K+1	ED	70
		TEDT(K)=T(JJ)	ED	71
		IEDT(K)=ITEMP(JJ)	ED	72
	6	CONTINUE	ED	73
		TMIN=TEDT(1)	ED	74
75		K1=K1-1	ED	75
		GO TO 8	ED	76
	7	TMIN=T(1)	ED	77
	8	JSAVE=1	ED	78
		DO 11 J=1,K1	ED	79
80		IF (I.EQ.1) TEDT(J)=T(J)	ED	80

LISTING OF DECK: EDIT

PAGE 3

CARD NO.

81		IF (I.EQ.1) IEDT(J)=ITEMP(J)	ED 81
		IF (TEDT(J).LE.TMIN) GO TO 9	ED 82
		GO TO 10	ED 83
	9	JSAVE=J	ED 84
85		TMIN=TEDT(J)	ED 85
	10	T(J)=TEDT(J)	ED 86
		ITEMP(J)=IEDT(J)	ED 87
	11	CONTINUE	ED 88
		ISAVE(I)=IEDT(JSAVE)	ED 89
90	12	TSAVE(I)=TMIN	ED 90
		DO 13 I=1,N	ED 91
		K1=ISAVE(I)	ED 92
	13	YSAVE(I)=Y(K1)	ED 93
	C		ED 94
95	C	EDIT DATA	ED 95
	C		ED 96
		N1=N-1	ED 97
		IF (N1.LT.1.OR.TOLR.LE.0.0) GO TO 31	ED 98
		K1=0	ED 99
100		DO 15 I=1,N1	ED 100
		ITEMP(I)=0	ED 101
		DIFF=ABS(TSAVE(I+1)-TSAVE(I))	ED 102
		IF (DIFF.LE.TOLR) GO TO 14	ED 103
		GO TO 15	ED 104
105	14	ITEMP(I)=I	ED 105
		K1=1	ED 106
	15	CONTINUE	ED 107
		IF (K1.EQ.0) GO TO 31	ED 108
		N=1	ED 109
110		I=1	ED 110
	16	K1=I	ED 111
		J=0	ED 112
		IF (I.GT.N1) GO TO 27	ED 113
		IF (ITEMP(I).NE.0) GO TO 17	ED 114
115		IEDT(N)=ISAVE(I)	ED 115
		N=N+1	ED 116
		I=I+1	ED 117
		IEDT(N)=ISAVE(I)	ED 118
		GO TO 16	ED 119
120	17	I=I+1	ED 120

CARD NO.

121		J=J+1	ED 121
		IF (I.EQ.IN) GO TO 21	ED 122
		IF (ITEMP(I).EQ.0) GO TO 18	ED 123
		GO TO 17	ED 124
125	18	K2=K1+J	ED 125
		IF (K1.EQ.1) GO TO 19	ED 126
		I1=K1-1	ED 127
		I2=K2+1	ED 128
		GO TO 23	ED 129
130	19	I1=K2	ED 130
	20	I1=I1+1	ED 131
		I2=I1+1	ED 132
		IF (I2.GT.IN) IFRR=1	ED 133
		IF (I2.GT.IN) GO TO 29	ED 134
135		DIFF=ABS(TSAVE(I2)-TSAVE(I1))	ED 135
		IF (DIFF.LE.TOLR) GO TO 20	ED 136
		GO TO 23	ED 137
	21	I2=K1	ED 138
	22	I2=I2-1	ED 139
140		I1=I2-1	ED 140
		IF (I1.LT.1) IERR=2	ED 141
		IF (I1.LT.1) GO TO 29	ED 142
		DIFF=ABS(TSAVE(I2)-TSAVE(I1))	ED 143
		IF (DIFF.LE.TOLR) GO TO 22	ED 144
145		K2=K1+J	ED 145
	23	IF (I1.LT.1) IERR=3	ED 146
		IF (I1.LT.1) GO TO 29	ED 147
		IF (I2.GT.IN) IERR=4	ED 148
		IF (I2.GT.IN) GO TO 29	ED 149
150		IEDT(N)=ISAVE(K1)	ED 150
		T(1)=YSAVE(I1)*TSAVE(I2)-YSAVE(I2)*TSAVE(I1)	ED 151
		T(2)=YSAVE(I2)-YSAVE(I1)	ED 152
		T(3)=TSAVE(I2)-TSAVE(I1)	ED 153
		TMIN=ABS(YSAVE(K1)-(T(1)+T(2)*TSAVE(K1))/T(3))	ED 154
155		K1=K1+1	ED 155
		DO 25 K=K1,K2	ED 156
		DIFF=ABS(YSAVE(K)-(T(1)+T(2)*TSAVE(K))/T(3))	ED 157
		IF (DIFF.LE.TMIN) GO TO 24	ED 158
		GO TO 25	ED 159
160	24	TMIN=DIFF	ED 160

CARD NO.

161		IEDT(N)=ISAVE(K)	ED 161
	25	CONTINUE	ED 162
		IF (K2.EQ.IN) GO TO 27	ED 163
		IF (I.EQ.N1) GO TO 26	ED 164
165		N=N+1	ED 165
		I=I+1	ED 166
		GO TO 16	ED 167
	26	N=N+1	ED 168
		IEDT(N)=ISAVE(IN)	ED 169
170	27	DO 28 K=1,N	ED 170
		K1=IEDT(K)	ED 171
	28	TEDT(K)=TIN(K1)	ED 172
		GO TO 33	ED 173
	29	IF (NSTORE.EQ.0) PRINT 40, IRUN,TOLR,IERR	ED 174
175		DO 30 I=1,IN	ED 175
		K1=ISAVE(I)	ED 176
		CALL PNTPLT (X(K1),Y(K1),NSYM,ISIZE)	ED 177
		IF (NSTORE.EQ.0) PRINT 41, I,K1,TSAVE(I),X(K1),Y(K1)	ED 178
	30	CONTINUE	ED 179
180		RETURN	ED 180
	31	DO 32 I=1,IN	ED 181
		K1=ISAVE(I)	ED 182
		TEDT(I)=TIN(K1)	ED 183
	32	IEDT(I)=K1	ED 184
185	C		ED 185
	C	WRITE EDITED DATA ON TAPE 7 FOR USE IN SUBROUTINE PRTEDT	ED 186
	C		ED 187
	33	IF (NSTORE.EQ.1) GO TO 34	ED 188
		WRITE (7) IRUN,IN,N,TOLR	ED 189
190		WRITE (7) (TIN(I),I=1,IN)	ED 190
		WRITE (7) (ISAVE(I),TSAVE(I),I=1,IN)	ED 191
		WRITE (7) (IEDT(I),TEDT(I),I=1,N)	ED 192
	C		ED 193
	C	REORDER T, X, AND Y FOR PLOTTING	ED 194
195	C		ED 195
	34	DO 35 I=1,IN	ED 196
	35	T(I)=TEDT(I)	ED 197
		DO 36 I=1,IN	ED 198
		TSAVE(I)=X(I)	ED 199
200	36	TEDT(I)=Y(I)	ED 200

CARD NO.

201		DO 37 I=1,IN	ED 201
		K=IEDT(I)	ED 202
		X(I)=TSAVE(K)	ED 203
	37	Y(I)=TEDT(K)	ED 204
205	C		ED 205
	C	PLOT DATA POINTS NOT EDITED	ED 206
	C		ED 207
		IF (N.EQ.IN) RETURN	ED 208
		DO 39 I=1,IN	ED 209
210		K1=0	ED 210
		DO 38 J=1,N	ED 211
		IF (IEDT(J).EQ.I) K1=1	ED 212
	38	CONTINUE	ED 213
		IF (K1.EQ.0) CALL PNTPLT (TSAVE(I),TEDT(I),NSYM,ISTZE)	ED 214
215		IF ((K1.EQ.0).AND.(NSTORE.EQ.0)) WRITE (7) I,TIN(I)	ED 215
	39	CONTINUE	ED 216
	C		ED 217
	C		ED 218
	C	RETURN TO CALLING PROGRAM	ED 219
220	C		ED 220
		RETURN	ED 221
	C		ED 222
	40	FORMAT (//2X,19HATTEMPT TO EDIT RUN,I7,1X,29HFAILED FOR AN EDIT TO	ED 223
		1LERANCE=,F8.3,10X,9HERROR NO.,I3//4X,1HI,4X,2HIS,6X,1HT,9X,1HX,9X,	ED 224
225		21HY//)	ED 225
	41	FORMAT (2I5,3F10.4)	ED 226
		END	ED 227-

CARD NO.

1		SUBROUTINE PRTEDT	PR	1
	C		PR	2
	C	ROUTINE TO PRINT EDITED DATA STORED ON TAPE 7	PR	3
	C		PR	4
5	C	CODED BY -- HARRY L. MORGAN	PR	5
	C	NASA/LARC/TAD/AAB	PR	6
	C	1983	PR	7
	C	WORK ARRAYS	PR	8
	C	COMMON /WORK/ TIN(50),ISAVE(50),TSAVE(50),IEDT(50),TEDT(50),IUN(50	PR	9
	C	1),TUN(50)	PR	10
10	C	END FILE 7	PR	11
		REWIND 7	PR	12
		PRINT 7	PR	13
	1	READ (7) IRUN,IN,N,TOLR	PR	14
15		IF (EOF(7)) 6,2	PR	15
	2	NR=IN-N	PR	16
		READ (7) (TIN(I),I=1,IN)	PR	17
		READ (7) (ISAVE(I),TSAVE(I),I=1,IN)	PR	18
		READ (7) (IEDT(I),TEDT(I),I=1,N)	PR	19
20		IF (NR.LE.0) GO TO 4	PR	20
		DO 3 I=1,NR	PR	21
	3	READ (7) IUN(I),TUN(I)	PR	22
	4	PRINT 8, IRUN,TOLR	PR	23
		DO 5 I=1,IN	PR	24
25		IF ((I.LE.N).AND.(I.LE.NR)) PRINT 9, I,TIN(I),ISAVE(I),TSAVE(I),IE	PR	25
		1DT(I),TEDT(I),IUN(I),TUN(I)	PR	26
		IF ((I.LE.N).AND.(I.GT.NR)) PRINT 10, I,TIN(I),ISAVE(I),TSAVE(I),I	PR	27
		1EDT(I),TEDT(I)	PR	28
		IF ((I.GT.N).AND.(I.LE.NR)) PRINT 11, I,TIN(I),ISAVE(I),TSAVE(I),I	PR	29
30		1UN(I),TUN(I)	PR	30
		IF ((I.GT.N).AND.(I.GT.NR)) PRINT 12, I,TIN(I),ISAVE(I),TSAVE(I)	PR	31
	5	CONTINUE	PR	32
		GO TO 1	PR	33
	6	RETURN	PR	34
35	C		PR	35
	7	FORMAT (1H1,10X,35HTHE FOLLOWING RUNS HAVE BEEN EDITED)	PR	36
	8	FORMAT (//1X,9HRUN NO. =,I7,10X,11HTOLERANCE =,F8.3//9X,5HINPUT,19	PR	37
		1X,6HSORTED,19X,6HEDITED,20X,7HDELETED//4(1X,9HPOINT NO.,3X,7HT-VAL	PR	38
		2UE,5X)//	PR	39
40	9	FORMAT (4(5X,T3,2X,F10.4,5X))	PR	40

LISTING OF DECK: PRTEDT

PAGE 2

CARD NO.

41

10 FORMAT (3(5X,I3,2X,F10.4,5X))
 11 FORMAT (2(5X,I3,2X,F10.4,5X),30X,I3,2X,F10.4)
 12 FORMAT (2(5X,I3,2X,F10.4,5X))
 END

PR 41
 PR 42
 PR 43
 PR 44-

LISTING OF DECK: ZERO LN

PAGE 1

CARD NO.

1		SUBROUTINE ZERO LN (XO,YO,DIST,LINE)	ZN	1
	C		ZN	2
	C	ROUTINE TO DRAW ZERO LINES	ZN	3
	C		ZN	4
5	C	CODED BY -- HARRY L. MORGAN	ZN	5
	C	NASA/LARC/TAD/AAB	ZN	6
	C	1983	ZN	7
	C	PARAMETER DEFINITION	ZN	8
	C	XO,YO - STARTING LOCATION OF ZERO LINE	ZN	9
	C	DIST - LENGTH OF ZERO LINE	ZN	10
10	C	LINE - AXIS OPTION	ZN	11
	C	LINE=1, ZERO LINE PARALLEL TO X-AXIS	ZN	12
	C	LINE=2, ZERO LINE PARALLEL TO Y-AXIS	ZN	13
	C		ZN	14
	C	NOTE - THICKNESS OF ZERO LINE = 2.*D	ZN	15
15		D=0.014	ZN	16
		CALL CALPLT (XO,YO,3)	ZN	17
		IF (LINE.EQ.2) GO TO 1	ZN	18
		X=XO+DIST	ZN	19
		CALL CALPLT (X,YO,2)	ZN	20
20		Y=YO+D	ZN	21
		CALL CALPLT (X,Y,2)	ZN	22
		CALL CALPLT (XO,Y,2)	ZN	23
		Y=YO-D	ZN	24
		CALL CALPLT (XO,Y,2)	ZN	25
25		CALL CALPLT (X,Y,2)	ZN	26
		CALL CALPLT (X,YO,2)	ZN	27
		GO TO 2	ZN	28
	1	Y=YO+DIST	ZN	29
		CALL CALPLT (XO,Y,2)	ZN	30
30		X=XO+D	ZN	31
		CALL CALPLT (X,Y,2)	ZN	32
		CALL CALPLT (X,YO,2)	ZN	33
		X=XO-D	ZN	34
		CALL CALPLT (X,YO,2)	ZN	35
35		CALL CALPLT (X,Y,2)	ZN	36
		CALL CALPLT (XO,Y,2)	ZN	37
	2	CALL CALPLT (XO,YO,3)	ZN	38
		RETURN	ZN	39-
		END		

CARD NO.

1		SUBROUTINE CURPLT (T,X,Y,N,IVAR,NSYM,ISIZE,IOP,IRUN,TENSION)	CU	1
	C		CU	2
	C	ROUTINE TO PLOT AND FAIR DATA	CU	3
	C		CU	4
5	C	CODED BY -- HARRY L. MORGAN	CU	5
	C	NASA/LARC/TAD/A4B	CU	6
	C	1983	CU	7
	C	PARAMETER DEFINITION	CU	8
	C	T - ARRAY CONTAINING INDEPENDENT VARIABLE	CU	9
	C	X - ARRAY CONTAINING X VARIABLE	CU	10
10	C	Y - ARRAY CONTAINING Y VARIABLE	CU	11
	C	N - NUMBER OF POINTS TO BE PLOTTED	CU	12
	C	IVAR - VARIABLE CODE IVAR=0 IF X AND T ARE THE SAME	CU	13
	C	IVAR=1 IF X AND T ARE DIFFERENT	CU	14
	C	NSYM - SYMBOL NUMBER	CU	15
15	C	ISIZE - SYMBOL SIZE	CU	16
	C	IOP - PLOTTING OPTION IOP=0 PLOT SYMBOLS ONLY	CU	17
	C	IOP=1 PLOT CURVE AND SYMBOLS	CU	18
	C	IRUN - RUN NUMBER	CU	19
	C	TENSION - SPLINE TENSION	CU	20
20	C	T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING ROUTINE	CU	21
	C	DIMENSION T(1), X(1), Y(1), H(3)	CU	22
	C	WORK ARRAYS	CU	23
	C	COMMON /WORK/ DS1(105),DS2(105),DUMX(105),DUMY(105)	CU	24
25	C	COMMON /PLT/ MX(50),MY(50),A(50)	CU	25
	C	REAL MX,MY	CU	26
	C	ROUTINE CONSTANTS	CU	27
	C	DATA NH/101/,XNT/100./,EPS/0.0001/,NTA/100/,H/.02,.03,.04/	CU	28
30	C	SELECT PLOTTING OPTION	CU	29
	C		CU	30
	C	IF (IOP.EQ.0.OR.N.LT.3) GO TO 1	CU	31
	C	GO TO 3	CU	32
35	C		CU	33
	C	PLOT SYMBOLS ONLY	CU	34
	C		CU	35
	1	DO 2 I=1,N	CU	36
	2	CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)	CU	37
40		RETURN	CU	38
			CU	39
			CU	40

CARD NO.

41	C		CU 41
	C	PLOT CURVE AND SYMBOL	CU 42
	C		CU 43
	C	CHECK TO SEE IF X OR T IS STRICTLY INCREASING	CU 44
45	3	DO 5 I=2,N	CU 45
		IF (IVAR.EQ.1) GO TO 4	CU 46
		IF (X(I).LT.X(I-1)) GO TO 6	CU 47
		GO TO 5	CU 48
	4	IF (T(I).LT.T(I-1)) GO TO 6	CU 49
50	5	CONTINUE	CU 50
		GO TO 7	CU 51
	C	PRINT ERROR MESSAGE IF X OR T NOT STRICTLY INCREASING	CU 52
	6	IF (IVAR.EQ.0) PRINT 26, IRUN	CU 53
		IF (IVAR.EQ.0) PRINT 27, (X(I),Y(I),I=1,N)	CU 54
55		IF (IVAR.EQ.1) PRINT 28, IRUN	CU 55
		IF (IVAR.EQ.1) PRINT 29, (T(I),X(I),Y(I),I=1,N)	CU 56
	C	PLOT SYMBOLS ONLY IF X OR T NOT STRICTLY INCREASING	CU 57
		GO TO 1	CU 58
	C	FIT SPLINE CURVE THROUGH DATA POINTS	CU 59
60	7	IF (IVAR.EQ.0) CALL CUBSPL (X,Y,N,MX,TENSION,TENS,A)	CU 60
		IF (IVAR.EQ.1) CALL CUBSPL (T,X,N,MY,TENSION,TENS,A)	CU 61
		IF (IVAR.EQ.1) CALL CUBSPL (T,Y,N,MY,TENSION,TENS,A)	CU 62
	C	PLOT FIRST POINT	CU 63
		CALL PNTPLT (X(1),Y(1),NSYM,ISIZE)	CU 64
65	C	PLOT AND FAIR REMAINING POINTS	CU 65
		NN=N-1	CU 66
		DO 25 I=1,NN	CU 67
	C	COMPUTE STRAIGHT LINE DISTANCE BETWEEN TWO POINTS. IF DISTANCE	CU 68
	C	LESS THAN SYMBOL DIAMETER, PLOT POINTS ONLY.	CU 69
70		X1=X(I+1)-X(I)	CU 70
		Y1=Y(I+1)-Y(I)	CU 71
		DS=SQRT(X1*X1+Y1*Y1)	CU 72
		T1=ATANF(Y1,X1)	CU 73
		CALL SYMBS (NSYM,ISIZE,XS1,YS1,T1)	CU 74
75		DSS1=SQRT(XS1*XS1+YS1*YS1)	CU 75
		X1=-X1	CU 76
		Y1=-Y1	CU 77
		T1=ATANF(Y1,X1)	CU 78
		CALL SYMBS (NSYM,ISIZE,XS2,YS2,T1)	CU 79
80		DSS2=SQRT(XS2*XS2+YS2*YS2)	CU 80

CARD NO.

81		IF ((DSS1+DSS2).GE.DS) GO TO 24	CU 81
	C	COMPUTE DISTANCE ALONG CURVE AS A FUNCTION OF X OR T BETWEEN	CU 82
	C	POINT I AND I+1	CU 83
		IF (IVAR.EQ.0) NT=IFIX(XNT*(X(I+1)-X(I)))+1	CU 84
85		IF (IVAR.EQ.1) NT=IFIX(XNT*(T(I+1)-T(I)))+1	CU 85
		IF (NT.LT.3) NT=3	CU 86
		IF (NT.GT.NH) NT=NH	CU 87
		IF (IVAR.EQ.0) DELTA=(X(I+1)-X(I))/FLOAT(NT-1)	CU 88
		IF (IVAR.EQ.1) DELTA=(T(I+1)-T(I))/FLOAT(NT-1)	CU 89
90		DUMX(1)=X(I)	CU 90
		DUMY(1)=Y(I)	CU 91
		DS1(1)=0.0	CU 92
		IF (IVAR.EQ.0) DA=X(I)	CU 93
		IF (IVAR.EQ.1) DA=T(I)	CU 94
95	C		CU 95
		DO 9 J=2,NT	CU 96
		DA=DA+DELTA	CU 97
		IF (IVAR.EQ.1) GO TO 8	CU 98
		DUMX(J)=DA	CU 99
100		DUMY(J)=FUNC(DA,X(I),X(I+1),Y(I),Y(I+1),MX(I),MX(I+1),TENS)	CU 100
		GO TO 9	CU 101
	8	DUMY(J)=FUNC(DA,T(I),T(I+1),Y(I),Y(I+1),MY(I),MY(I+1),TENS)	CU 102
		DUMX(J)=FUNC(DA,T(I),T(I+1),X(I),X(I+1),MX(I),MX(I+1),TENS)	CU 103
	9	DS1(J)=SQRT((DUMX(J)-X(I))**2+(DUMY(J)-Y(I))**2)	CU 104
105	C		CU 105
		DO 10 J=1,NT	CU 106
		K=NT+1-J	CU 107
	10	DS2(J)=SQRT((DUMX(K)-X(I+1))**2+(DUMY(K)-Y(I+1))**2)	CU 108
	C	FIND X AND Y LOCATION WHERE SYMBOL AND CURVE INTERSECT	CU 109
110		DELTA=H(ISIZE)	CU 110
		IF (IVAR.EQ.0) DA=X(I)	CU 111
		IF (IVAR.EQ.1) DA=T(I)	CU 112
	C		CU 113
		DO 14 J=2,NTA	CU 114
115		DA=DA+DELTA	CU 115
		IF (IVAR.EQ.1) GO TO 11	CU 116
		IF (DA.GE.X(I+1)) GO TO 15	CU 117
		X1=DA-X(I)	CU 118
		Y1=FUNC(DA,X(I),X(I+1),Y(I),Y(I+1),MX(I),MX(I+1),TENS)-Y(I)	CU 119
120		GO TO 12	CU 120

CARD NO.

121	11	IF (DA.GE.T(I+1)) GO TO 15	CU 121
		Y1=FUNC(DA,T(I),T(I+1),Y(I),Y(I+1),MY(I),MY(I+1),TENS)-Y(I)	CU 122
		X1=FUNC(DA,T(I),T(I+1),X(I),X(I+1),MX(I),MX(I+1),TENS)-X(I)	CU 123
	12	DS=SQRT(X1*X1+Y1*Y1)	CU 124
125		T1=ATANF(Y1,X1)	CU 125
		CALL SYMBS (NSYM,ISIZE,XS1,YS1,T1)	CU 126
		DSS1=SQRT(XS1*XS1+YS1*YS1)	CU 127
		IF (ABS(DS-DSS1).LE.EPS) GO TO 15	CU 128
		IF (DS.GT.DSS1) GO TO 13	CU 129
130		GO TO 14	CU 130
	13	DA=DA-DELTA	CU 131
		DELTA=DELTA/2.	CU 132
	14	CONTINUE	CU 133
	C		CU 134
135	15	XS1=XS1+X(I)	CU 135
		YS1=YS1+Y(I)	CU 136
		DELTA=H(ISIZE)	CU 137
		IF (IVAR.EQ.0) DA=X(I+1)	CU 138
		IF (IVAR.EQ.1) DA=T(I+1)	CU 139
140	C		CU 140
		DO 19 J=2,NTA	CU 141
		DA=DA-DELTA	CU 142
		IF (IVAR.EQ.1) GO TO 16	CU 143
		IF (DA.LE.X(I)) GO TO 20	CU 144
145		X1=DA-X(I+1)	CU 145
		Y1=FUNC(DA,X(I),X(I+1),Y(I),Y(I+1),MX(I),MX(I+1),TENS)-Y(I+1)	CU 146
		GO TO 17	CU 147
	16	IF (DA.LE.T(I)) GO TO 20	CU 148
		Y1=FUNC(DA,T(I),T(I+1),Y(I),Y(I+1),MY(I),MY(I+1),TENS)-Y(I+1)	CU 149
150		X1=FUNC(DA,T(I),T(I+1),X(I),X(I+1),MX(I),MX(I+1),TENS)-X(I+1)	CU 150
	17	DS=SQRT(X1*X1+Y1*Y1)	CU 151
		T1=ATANF(Y1,X1)	CU 152
		CALL SYMBS (NSYM,ISIZE,XS2,YS2,T1)	CU 153
		DSS2=SQRT(XS2*XS2+YS2*YS2)	CU 154
155		IF (ABS(DS-DSS2).LE.EPS) GO TO 20	CU 155
		IF (DS.GT.DSS2) GO TO 18	CU 156
		GO TO 19	CU 157
	18	DA=DA+DELTA	CU 158
		DELTA=DELTA/2.	CU 159
160	19	CONTINUE	CU 160

CARD NO.

161	C		CU 161
	20	XS2=XS2+X(I+1)	CU 162
		YS2=YS2+Y(I+1)	CU 163
		NP=1	CU 164
165		NDS1=0	CU 165
	C		CU 166
		DO 21 J=2,NT	CU 167
		IF ((DS1(J).LT.DSS1).AND.(NDS1.EQ.0)) GO TO 21	CU 168
		IF ((DS1(J).GE.DSS1).AND.(NDS1.EQ.0)) NDS1=1	CU 169
170		NP=NP+1	CU 170
		DUMX(NP)=DUMX(J)	CU 171
		DUMY(NP)=DUMY(J)	CU 172
	21	CONTINUE	CU 173
	C	PLOT CURVE BETWEEN POINTS	CU 174
175		DUMX(1)=XS1	CU 175
		DUMY(1)=YS1	CU 176
		DO 22 J=1,NT	CU 177
		IF (DS2(J).LE.DSS2) GO TO 22	CU 178
		GO TO 23	CU 179
180	22	NP=NP-1	CU 180
	23	NP=NP+1	CU 181
		DUMX(NP)=XS2	CU 182
		DUMY(NP)=YS2	CU 183
		DUMX(NP+1)=DUMY(NP+1)=0.0	CU 184
185		DUMX(NP+2)=DUMY(NP+2)=1.0	CU 185
		CALL LINE (DUMX,DUMY,NP,1,0,0,0)	CU 186
	C	PLOT SYMBOL AT POINT I+1	CU 187
	24	CALL PNTPLT (X(I+1),Y(I+1),NSYM,ISIZE)	CU 188
	C	ADVANCE TO NEXT POINT	CU 189
190	25	CONTINUE	CU 190
	C		CU 191
	C	RETURN TO CALLING PROGRAM	CU 192
	C		CU 193
		RETURN	CU 194
195	C		CU 195
	26	FORMAT (1H1//5X,36HX IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HX	CU 196
		1,13X,1HY//)	CU 197
	27	FORMAT (5X,2F15.4)	CU 198
	28	FORMAT (1H1//5X,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT	CU 199
200		1,13X,1HX,13X,1HY//)	CU 200

LISTING OF DECK: CURPLT

PAGE 6

CARD NO.

201

29

FORMAT (5X,3F15.4)
END

CU 201
CU 202-

CARD NO.

1		FUNCTION ATANF (DY,DX)	AF	1
	C		AF	2
	C	ROUTINE TO COMPUTE ARCTANGENT FOR ANGLES FROM 0 TO 360 DEGREES	AF	3
	C		AF	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAR 1983	AF	5
	C		AF	6
		IF (DX.EQ.0.0) GO TO 1	AF	7
		ATANF=ABS(DY/DX)	AF	8
		ATANF=ATAN(ATANF)*57.2957795131	AF	9
10		IF ((DX.GT.0.0).AND.(DY.LT.0.0)) ATANF=360.-ATANF	AF	10
		IF ((DX.LT.0.0).AND.(DY.GT.0.0)) ATANF=180.-ATANF	AF	11
		IF ((DX.LT.0.0).AND.(DY.LT.0.0)) ATANF=180.+ATANF	AF	12
		RETURN	AF	13
	1	IF (DY.LT.0.0) ATANF=270.	AF	14
15		IF (DY.GE.0.0) ATANF=90.	AF	15
		RETURN	AF	16
		END	AF	17-

LISTING OF DECK: SYMBS

PAGE 1

CARD NO.

1		SUBROUTINE SYMBS (NO,IS,X,Y,T)	SY	1
	C		SY	2
	C	ROUTINE TO COMPUTE THE X AND Y INTERCEPTS OF A RADIAL LINE DRAWN	SY	3
	C	FROM THE CENTER OF A GIVEN SYMBOL WITH THE SIDE OF THAT SYMBOL	SY	4
5	C		SY	5
	C	CODED BY -- HARRY L. MORGAN	SY	6
	C	NASA/LARC/TAD/AAB 1983	SY	7
	C	PARAMETER DEFINITION	SY	8
	C	NO - SYMBOL NUMBER	SY	9
10	C	IS - SYMBOL SIZE 1-SMALL 2-MEDIUM 3-LARGE	SY	10
	C	X AND Y - INTERCEPT OF SYMBOL AND RADIAL DRAWN FROM SYMBOL	SY	11
	C	CENTER AT ANGLE T.	SY	12
	C	T - ANGLE OF RADIAL DRAWN FROM SYMBOL CENTER	SY	13
	C		SY	14
15		DIMENSION SCALE(3)	SY	15
		DATA RAD/57.2957795131/,PI/3.141592654/	SY	16
		DATA DA/1.4142135624/,R/1.7320508076/,SCALE/.13,.16,.19/	SY	17
		DATA T1/213.6900675260/,T2/326.3099324740/,T3/116.5650511771/,T4/3	SY	18
		133.4349488229/,T7/11.3099324740/,T8/168.6900675260/,T9/218.6598082	SY	19
20		2541/,S1/321.3401917459/,S2/185.7105931375/,S3/354.2894068625/,S4/1	SY	20
		31.3099324740/,S5/168.6900675260/,S6/218.6598082541/,S7/321.3401917	SY	21
		4459/	SY	22
	C		SY	23
		IF ((NO.EQ.1).OR.(NO.EQ.11)) GO TO 1	SY	24
25		IF ((NO.EQ.2).OR.(NO.EQ.12)) GO TO 2	SY	25
		IF ((NO.EQ.3).OR.(NO.EQ.13)) GO TO 8	SY	26
		IF ((NO.EQ.4).OR.(NO.EQ.14)) GO TO 13	SY	27
		IF ((NO.EQ.5).OR.(NO.EQ.15)) GO TO 17	SY	28
		IF ((NO.EQ.6).OR.(NO.EQ.16)) GO TO 21	SY	29
30		IF ((NO.EQ.7).OR.(NO.EQ.17)) GO TO 26	SY	30
		IF ((NO.EQ.8).OR.(NO.EQ.18)) GO TO 31	SY	31
		IF ((NO.EQ.9).OR.(NO.EQ.19)) GO TO 35	SY	32
		IF ((NO.EQ.10).OR.(NO.EQ.20)) GO TO 40	SY	33
		IF ((NO.EQ.21).OR.(NO.EQ.22)) GO TO 45	SY	34
35		IF (NO.GT.22) RETURN	SY	35
	C		SY	36
	C	SYMBOL NUMBER 1 OR 11	SY	37
	C		SY	38
	1	X=SCALE(IS)*.5525*COS(T/RAD)	SY	39
40		Y=SCALE(IS)*.5525*SIN(T/RAD)	SY	40

CARD NO.

41		RETURN	SY 41
	C		SY 42
	C	SYMBOL NUMBER 2 OR 12	SY 43
	C		SY 44
45	2	IF ((T.GE.0.).AND.(T.LT.45.)) GO TO 3	SY 45
		IF ((T.GE.45.0.).AND.(T.LT.135.)) GO TO 4	SY 46
		IF ((T.GE.135.).AND.(T.LT.225.)) GO TO 5	SY 47
		IF ((T.GE.225.).AND.(T.LT.315.)) GO TO 6	SY 48
		IF ((T.GE.315.).AND.(T.LE.360.)) GO TO 7	SY 49
50	3	X=SCALE(IS)/2.	SY 50
		Y=X*TAN(T/RAD)	SY 51
		RETURN	SY 52
	4	Y=SCALE(IS)/2.	SY 53
		IF (T.EQ.90.) X=0.0	SY 54
55		IF (T.NE.90.) X=Y/TAN(T/RAD)	SY 55
		RETURN	SY 56
	5	X=-SCALE(IS)/2.	SY 57
		Y=X*TAN(T/RAD)	SY 58
		RETURN	SY 59
60	6	Y=-SCALE(IS)/2.	SY 60
		IF (T.EQ.270.) X=0.0	SY 61
		IF (T.NE.270.) X=Y/TAN(T/RAD)	SY 62
		RETURN	SY 63
	7	X=SCALE(IS)/2.	SY 64
65		Y=X*TAN(T/RAD)	SY 65
		RETURN	SY 66
	C		SY 67
	C	SYMBOL NUMBER 3 OR 13	SY 68
	C		SY 69
70	8	IF ((T.GE.0.).AND.(T.LT.90.)) GO TO 9	SY 70
		IF ((T.GE.90.).AND.(T.LT.180.)) GO TO 10	SY 71
		IF ((T.GE.180.).AND.(T.LT.270.)) GO TO 11	SY 72
		IF ((T.GE.270.).AND.(T.LE.360.)) GO TO 12	SY 73
	9	X=SCALE(IS)*DA/2./(TAN(T/RAD)+1.)	SY 74
75		Y=-X+SCALE(IS)*DA/2.	SY 75
		RETURN	SY 76
	10	IF (T.EQ.90.) X=0.0	SY 77
		IF (T.NE.90.) X=SCALE(IS)*DA/2./(TAN(T/RAD)-1.)	SY 78
		Y=X+SCALE(IS)*DA/2.	SY 79
80		RETURN	SY 80

LISTING OF DECK: SYMBS

PAGE 3

CARD NO.

81	11	X=-SCALE(IS)*DA/2./((TAN(T/RAD)+1.)	SY 81
		Y=-X-SCALE(IS)*DA/2.	SY 82
		RETURN	SY 83
	12	IF (T.EQ.270.) X=0.0	SY 84
85		IF (T.NE.270.) X=-SCALE(IS)*DA/2./((TAN(T/RAD)-1.)	SY 85
		Y=X-SCALE(IS)*DA/2.	SY 86
		RETURN	SY 87
	C		SY 88
	C	SYMBOL NUMBER 4 OR 14	SY 89
90	C		SY 90
	13	IF ((T.GE.0.).AND.(T.LT.90.)) GO TO 14	SY 91
		IF ((T.GE.90.).AND.(T.LT.T1)) GO TO 15	SY 92
		IF ((T.GE.T1).AND.(T.LT.T2)) GO TO 16	SY 93
	14	X=(2./3.)*SCALE(IS)*1.105/(TAN(T/RAD)+2.)	SY 94
95		Y=-2.*X+2.*SCALE(IS)*1.105/3.	SY 95
		RETURN	SY 96
	15	IF (T.EQ.90.) X=0.0	SY 97
		IF (T.NE.90.) X=(2./3.)*SCALE(IS)*1.105/(TAN(T/RAD)-2.)	SY 98
		Y=2.*X+2.*SCALE(IS)*1.105/3.	SY 99
100		RETURN	SY 100
	16	Y=-SCALE(IS)*1.105/3.	SY 101
		IF (T.NE.270.) X=0.0	SY 102
		IF (T.NE.270.) X=Y/TAN(T/RAD)	SY 103
		RETURN	SY 104
105	C		SY 105
	C	SYMBOL NUMBER 5 OR 15	SY 106
	C		SY 107
	17	IF ((T.GE.0.).AND.(T.LT.T3)) GO TO 18	SY 108
		IF ((T.GE.T3).AND.(T.LT.225.)) GO TO 19	SY 109
110		IF ((T.GE.225.).AND.(T.LT.T4)) GO TO 20	SY 110
	18	IF (T.EQ.90.) X=0.0	SY 111
		IF (T.NE.90.) X=SCALE(IS)*1.22222/3./((TAN(T/RAD)+1.)	SY 112
		Y=-X+SCALE(IS)*1.22222/3.	SY 113
		RETURN	SY 114
115	19	X=-SCALE(IS)*1.22222/3.	SY 115
		Y=X*TAN(T/RAD)	SY 116
		RETURN	SY 117
	20	Y=-SCALE(IS)*1.22222/3.	SY 118
		IF (T.EQ.270.) X=0.0	SY 119
120		IF (T.NE.270.) X=Y/TAN(T/RAD)	SY 120

CARD NO.

121		RETURN	SY 121
	C		SY 122
	C	SYMBOL NUMBER 6 OR 16	SY 123
	C		SY 124
125	21	A=4.*SCALE(IS)*1.22222/(3.*PI)	SY 125
		B=SCALE(IS)*1.22222-A	SY 126
		T5=ATAN(A/B)*RAD	SY 127
		T6=360.-T5	SY 128
		T5=90.+T5	SY 129
130		IF ((T.GE.0.).AND.(T.LT.T5)) GO TO 22	SY 130
		IF ((T.GE.T5).AND.(T.LT.225.)) GO TO 24	SY 131
		IF ((T.GE.225.).AND.(T.LT.T6)) GO TO 25	SY 132
	22	IF (T.EQ.90.) GO TO 23	SY 133
		BB=-2.*A*(1.+TAN(T/RAD))	SY 134
135		AA=TAN(T/RAD)**2+1.	SY 135
		CC=2.*A*A-(SCALE(IS)*1.22222)**2	SY 136
		X=SQRT(BB*BB-4.*AA*CC)/(2.*AA)	SY 137
		IF ((T.GE.0.).AND.(T.LT.90.)) X=X+BB/(2.*AA)	SY 138
		IF ((T.GE.90.).AND.(T.LT.180.)) X=-X+BB/(2.*AA)	SY 139
140		IF ((T.GE.270.).AND.(T.LE.360.)) X=X+BB/(2.*AA)	SY 140
		Y=X*TAN(T/RAD)	SY 141
		RETURN	SY 142
	23	X=0.0	SY 143
		Y=-A+SQRT((SCALE(IS)*1.22222)**2-A*A)	SY 144
145		RETURN	SY 145
	24	X=-A	SY 146
		Y=X*TAN(T/RAD)	SY 147
		RETURN	SY 148
	25	Y=-A	SY 149
150		IF (T.EQ.270.) X=0.0	SY 150
		IF (T.NE.270.) X=Y/TAN(T/RAD)	SY 151
		RETURN	SY 152
	C		SY 153
	C	SYMBOL NUMBER 7 OR 17	SY 154
155	C		SY 155
	26	IF ((T.GE.0.0).AND.(T.LT.T7)) GO TO 27	SY 156
		IF ((T.GE.T7).AND.(T.LT.T8)) GO TO 28	SY 157
		IF ((T.GE.T8).AND.(T.LT.T9)) GO TO 29	SY 158
		IF ((T.GE.T9).AND.(T.LT.S1)) GO TO 30	SY 159
160	27	X=SCALE(IS)/2.	SY 160

CARD NO.

161		Y=X*TAN(T/RAD)	SY 161
		RETURN	SY 162
	28	X=SCALE(IS)*.1*SIN(T/RAD)*COS(T/RAD)+SCALE(IS)*COS(T/RAD)*SORT(.25	SY 163
		1-(.1*COS(T/RAD))**2)	SY 164
165		Y=X*TAN(T/RAD)	SY 165
		RETURN	SY 166
	29	X=-SCALE(IS)/2.	SY 167
		Y=X*TAN(T/RAD)	SY 168
		RETURN	SY 169
170	30	Y=-SCALE(IS)*.4	SY 170
		IF (T.EQ.270.) X=0.0	SY 171
		IF (T.NE.270.) X=Y/TAN(T/RAD)	SY 172
		RETURN	SY 173
	C		SY 174
175	C	SYMBOL NUMBER 8 OR 18	SY 175
	C		SY 176
	31	IF ((T.GE.0.0).AND.(T.LT.S2)) GO TO 32	SY 177
		IF ((T.GE.S2).AND.(T.LE.270.)) GO TO 33	SY 178
		IF ((T.GT.270.).AND.(T.LE.S3)) GO TO 34	SY 179
180	32	X=-.05*SCALE(IS)*SIN(T/RAD)*COS(T/RAD)+SCALE(IS)*COS(T/RAD)*SORT(.125-	SY 180
		(.05*COS(T/RAD))**2)	SY 181
		Y=X*TAN(T/RAD)	SY 182
		RETURN	SY 183
	33	IF (T.EQ.270.) X=0.0	SY 184
185		IF (T.EQ.270.) Y=-SCALE(IS)*.55	SY 185
		IF (T.NE.270.) X=-SCALE(IS)*.55/(TAN(T/RAD)+1.)	SY 186
		IF (T.NE.270.) Y=X*TAN(T/RAD)	SY 187
		RETURN	SY 188
	34	X=-SCALE(IS)*.55/(TAN(T/RAD)-1.)	SY 189
190		Y=X*TAN(T/RAD)	SY 190
		RETURN	SY 191
	C		SY 192
	C	SYMBOL NUMBER 9 OR 19	SY 193
	C		SY 194
195	35	IF ((T.GE.0.0).AND.(T.LT.90.)) GO TO 36	SY 195
		IF ((T.GE.90.).AND.(T.LT.180.)) GO TO 37	SY 196
		IF ((T.GE.180.).AND.(T.LT.270.)) GO TO 38	SY 197
		IF ((T.GE.270.).AND.(T.LE.360.)) GO TO 39	SY 198
	36	X=SCALE(IS)*R/2./(TAN(T/RAD)+R)	SY 199
200		Y=X*TAN(T/RAD)	SY 200

CARD NO.

201		RETURN	SY 201
	37	IF (T.EQ.90.) X=0.0	SY 202
		IF (T.NE.90.) X=SCALE(IS)*R/2./(TAN(T/RAD)-R)	SY 203
		IF (T.EQ.90.) Y=SCALE(IS)*R/2.	SY 204
205		IF (T.NE.90.) Y=X*TAN(T/RAD)	SY 205
		RETURN	SY 206
	38	X=-SCALE(IS)*R/2./(TAN(T/RAD)+R)	SY 207
		Y=X*TAN(T/RAD)	SY 208
		RETURN	SY 209
210	39	IF (T.EQ.270.) X=0.0	SY 210
		IF (T.EQ.270.) Y=-SCALE(IS)*R/2.	SY 211
		IF (T.NE.270.) X=-SCALE(IS)*R/2./(TAN(T/RAD)-R)	SY 212
		IF (T.NE.270.) Y=X*TAN(T/RAD)	SY 213
		RETURN	SY 214
215	C		SY 215
	C	SYMBOL NUMBER 10 OR 20	SY 216
	C		SY 217
	40	IF ((T.GE.0.0).AND.(T.LT.S4)) GO TO 41	SY 218
		IF ((T.GE.S4).AND.(T.LT.S5)) GO TO 42	SY 219
220		IF ((T.GE.S5).AND.(T.LT.S6)) GO TO 43	SY 220
		IF ((T.GE.S6).AND.(T.LT.S7)) GO TO 44	SY 221
	41	X=SCALE(IS)/2.	SY 222
		Y=X*TAN(T/RAD)	SY 223
		RETURN	SY 224
225	42	IF (T.LT.90.) X=SCALE(IS)*.6/(TAN(T/RAD)+1.)	SY 225
		IF (T.EQ.90.) X=0.0	SY 226
		IF (T.GT.90.) X=SCALE(IS)*.6/(TAN(T/RAD)-1.)	SY 227
		IF (T.EQ.90.) Y=SCALE(IS)*.6	SY 228
		IF (T.NE.90.) Y=X*TAN(T/RAD)	SY 229
230		RETURN	SY 230
	43	X=-SCALE(IS)/2.	SY 231
		Y=X*TAN(T/RAD)	SY 232
		RETURN	SY 233
	44	Y=-SCALE(IS)*.4	SY 234
235		IF (T.EQ.270.) X=0.0	SY 235
		IF (T.NE.270.) X=Y/TAN(T/RAD)	SY 236
		RETURN	SY 237
	C		SY 238
	C	SYMBOL NUMBER 21 OR 22	SY 239
240	C		SY 240

LISTING OF DECK: SYMBS

PAGE 7

CARD NO.

241

45

X=.1*SCALE(IS)*COS(T/RAD)
Y=.1*SCALE(IS)*SIN(T/RAD)
RETURN
END

SY 241
SY 242
SY 243
SY 244-

CARD NO.

1		SUBROUTINE CURSPL (X,Y,N,YPP,TENSION,TENS,A)	CB	1
	C		CB	2
	C	THIS SUBROUTINE FITS A CUBIC SPLINE TO A SET OF Y VS X INPUT	CB	3
	C	POINTS	CB	4
5	C		CB	5
	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	CB	6
	C		CB	7
	C	IN CALLING PROGRAM DIMENSION X, Y, YPP, AND A BY N	CB	8
	C		CB	9
10		DIMENSION X(N), Y(N), YPP(N), A(N)	CB	10
	C		CB	11
	C	COMPUTE TENSION PARAMETER	CB	12
	C		CB	13
		TENS=TENSION*FLOAT(N-1)/(X(N)-X(1))	CB	14
15		IF (TENS.LT.0.0) TENS=0.0	CB	15
	C		CB	16
	C	COMPUTE SECOND DERIVATIVE AT END POINTS BY FITTING	CB	17
	C	Y=A*X**2+B*X+C TO THE LAST THREE POINTS AND SOLVE FOR A.	CB	18
	C	SECOND DERIVATIVE AT END POINT IS THEN EQUAL TO 2.*A	CB	19
20	C		CB	20
		H1=X(2)-X(3)	CB	21
		H2=X(3)-X(1)	CB	22
		H3=X(1)-X(2)	CB	23
		YPP(1)=2.*(Y(1)*H1+Y(2)*H2+Y(3)*H3)/(H1*X(1)**2+H2*X(2)**2+H3*X(3)	CB	24
25		1**2)	CB	25
		H1=X(N-1)-X(N)	CB	26
		H2=X(N)-X(N-2)	CB	27
		H3=X(N-2)-X(N-1)	CB	28
		YPP(N)=2.*(Y(N-2)*H1+Y(N-1)*H2+Y(N)*H3)/(H1*X(N-2)**2+H2*X(N-1)**2	CB	29
30		1+H3*X(N)**2)	CB	30
	C		CB	31
	C	PERFORM FORWARD ELIMINATION	CB	32
	C		CB	33
		N1=N-1	CB	34
35		H1=X(2)-X(1)	CB	35
		IF (TENS.NE.0.0) GO TO 1	CB	36
		CON=6.	CB	37
		E=H1	CB	38
		GO TO 2	CB	39
40	1	CON=TENS*TENS	CB	40

LISTING OF DECK: CUBSPL

PAGE 2

CARD NO.

41		H3=TENS/SINH(TENS*H1)	CB 41
		CTH=H3*COSH(TENS*H1)	CB 42
		E=1./H1-H3	CB 43
		DP=CTH-1./H1	CB 44
45	2	A(1)=0.0	CB 45
		DO 5 I=2,N1	CB 46
		H2=X(I+1)-X(I)	CB 47
		C=CON*((Y(I+1)-Y(I))/H2-(Y(I)-Y(I-1))/H1)	CB 48
		IF (TENS.NE.0.0) GO TO 3	CB 49
50		F=H2	CB 50
		D=H1*(2.-A(I-1))+2.*H2	CB 51
		GO TO 4	CB 52
	3	H3=TENS/SINH(TENS*H2)	CB 53
		CTH=H3*COSH(TENS*H2)	CB 54
55		F=1./H2-H3	CB 55
		DN=CTH-1./H2	CB 56
		D=DN+DP-E*A(I-1)	CB 57
		DP=DN	CB 58
	4	A(I)=F/D	CB 59
60		YPP(I)=(C-E*YPP(I-1))/D	CB 60
		E=F	CB 61
	5	H1=H2	CB 62
	C		CB 63
	C	PERFORM BACK SUBSTITUTION	CB 64
65	C		CB 65
		J=N	CB 66
		DO 6 I=2,N1	CB 67
		J=J-1	CB 68
	6	YPP(J)=YPP(J)-A(J)*YPP(J+1)	CB 69
70	C		CB 70
	C	RETURN TO CALLING PROGRAM	CB 71
	C		CB 72
		RETURN	CB 73
		END	CB 74-

LISTING OF DECK: FUNC

PAGE 1

CARD NO.

1		FUNCTION FUNC (X,X1,X2,Y1,Y2,YPP1,YPP2,TENS)	FC	1
	C		FC	2
	C	ROUTINE TO COMPUTE Y-VALUE AT A GIVEN X-VALUE ALONG SPLINE CURVE	FC	3
	C		FC	4
5	C	CODED BY -- HARRY L. MORGAN	FC	5
	C	NASA/LARC/TAD/AAB 1983	FC	6
	C		FC	7
		DX1=X2-X	FC	8
		DX2=X-X1	FC	9
		DH=X2-X1	FC	10
10		IF (TENS.EQ.0.0) GO TO 1	FC	11
		F1=YPP1/(TENS*TENS)	FC	12
		F2=YPP2/(TENS*TENS)	FC	13
		FUNC=(F1*SINH(TENS*DX1)+F2*SINH(TENS*DX2))/SINH(TENS*DH)+((Y1-F1)*	FC	14
		10X1+(Y2-F2)*DX2)/DH	FC	15
15		RETURN	FC	16
	1	FUNC=YPP1*DX1**3/(6.*DH)+YPP2*DX2**3/(6.*DH)+(Y1-YPP1*DH*DH/6.)*DX	FC	17
		11/DH+(Y2-YPP2*DH*DH/6.)*DX2/DH	FC	18
		RETURN	FC	19-
		END		

CARD NO.

1		SUBROUTINE DASHLN (T,X,Y,N,NSYM,IOP,ISIZE,IRUN,TENSION)	DH	1
	C		DH	2
	C	ROUTINE TO FAIR DATA WITH DASH LINES	DH	3
	C		DH	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	DH	5
	C		DH	6
	C	PARAMETER DEFINITION	DH	7
	C	T - ARRAY CONTAINING INDEPENDENT VARIABLE	DH	8
	C	X - ARRAY CONTAINING X VARIABLE	DH	9
10	C	Y - ARRAY CONTAINING Y VARIABLE	DH	10
	C	N - NUMBER OF POINTS TO BE PLOTTED	DH	11
	C	NSYM - LINE OR SYMBOL NUMBER	DH	12
	C	IOP - PLOTTING OPTION IOP=0 PLOT SYMBOLS ONLY	DH	13
	C	IOP=1 PLOT DASH LINE	DH	14
15	C	ISIZE - SYMBOL SIZE	DH	15
	C	IRUN - RUN NUMBER	DH	16
	C	TENSION - SPLINE TENSION	DH	17
	C		DH	18
	C	T, X, AND Y MUST BE PROPERLY DIMENSIONED IN CALLING ROUTINE	DH	19
20	C	DIMENSION T(1), X(1), Y(1), NL(22), NS(22)	DH	20
	C	WORK ARRAYS	DH	21
	C	COMMON /WORK/ S(166),XI(166),YI(166)	DH	22
	C	COMMON /PLT/ MX(50),MY(50),A(50)	DH	23
	C		DH	24
25	C	REAL MX,MY	DH	25
	C	ROUTINE CONSTANTS	DH	26
	C	NL = NUMBER OF LONG DASHES AND NS = NUMBER OF SHORT DASHES	DH	27
	C	DATA NL/1,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5,5/	DH	28
	C	DATA NS/0,1,1,2,3,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,5/	DH	29
30	C		DH	30
	C	ROUTINE CONSTANTS	DH	31
	C		DH	32
	C	NP=166	DH	33
	C	EPS=.02	DH	34
35	C	SL=.3	DH	35
	C	IF (NSYM.EQ.2) SL=.2	DH	36
	C	SS=.1	DH	37
	C	SP=.1	DH	38
	C		DH	39
40	C	SELECT PLOTTING OPTION	DH	40

CARD NO.

41	C	IF (IOP.EQ.0.OR.N.LT.3) GO TO 1	DH	41
		GO TO 3	DH	42
	C		DH	43
45	C	PLOT SYMBOLS ONLY	DH	44
	C		DH	45
	1	DO 2 I=1,N	DH	46
	2	CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)	DH	47
		RETURN	DH	48
50	C		DH	49
	C	PLOT DASH LINE	DH	50
	C		DH	51
	C	CHECK TO SEE IF T IS STRICTLY INCREASING	DH	52
	3	DO 4 I=2,N	DH	53
55		IF (T(I).LT.T(I-1)) GO TO 5	DH	54
	4	CONTINUE	DH	55
		GO TO 6	DH	56
	C	PRINT ERROR MESSAGE IF T IS NOT STRICTLY INCREASING	DH	57
	5	PRINT 15, IRUN	DH	58
60		PRINT 16, (T(I),X(I),Y(I),I=1,N)	DH	59
	C	IF T IS NOT STRICTLY INCREASING, PLOT SYMBOLS ONLY	DH	60
		GO TO 1	DH	61
	C	FIT SPLINE CURVE THROUGH DATA POINTS	DH	62
	6	CALL CUBSPL (T,X,N,MX,TENSION,TENS,A)	DH	63
65		CALL CUBSPL (T,Y,N,MY,TENSION,TENS,A)	DH	64
	C	COMPUTE NP EQUALLY SPACED POINTS ALONG CURVE	DH	65
		IF (NSYM.EQ.1) NP=NP-2	DH	66
		DT=(T(N)-T(1))/FLOAT(NP-1)	DH	67
		TI=T(1)	DH	68
70		S(1)=0.	DH	69
		M=2	DH	70
		DO 9 I=1,NP	DH	71
		IF (TI.LT.T(1)) TI=T(1)	DH	72
		IF (TI.GT.T(N)) TI=T(N)	DH	73
75		DO 7 J=M,N	DH	74
		K=J-1	DH	75
		IF (TI.GE.T(K).AND.TI.LE.T(J)) GO TO 8	DH	76
	7	CONTINUE	DH	77
	8	M=K	DH	78
80		IF (M.LE.1) M=2	DH	79
			DH	80

LISTING OF DECK: DASHLN

PAGE 3

CARD NO.

81		XI(I)=FUNC(TI,T(K),T(K+1),X(K),X(K+1),MX(K),MX(K+1),TENS)	DH 81
		YI(I)=FUNC(TI,T(K),T(K+1),Y(K),Y(K+1),MY(K),MY(K+1),TENS)	DH 82
		IF (I.EQ.1.OR.NSYM.EQ.1) GO TO 9	DH 83
		S(I)=S(I-1)+SORT((XI(I)-XI(I-1))**2+(YI(I)-YI(I-1))**2)	DH 84
85	9	TI=TI+DT	DH 85
		ST=S(NP)	DH 86
	C	IF NSYM=1, PLOT CONTINUOUS LINE	DH 87
		IF (NSYM.EQ.1) GO TO 13	DH 88
	C	COMPUTE LENGTH OF LONG AND SHORT DASH AND SPACE BETWEEN EACH	DH 89
90		NLG=NL(NSYM)	DH 90
		NSG=NS(NSYM)	DH 91
		DT=FLOAT(NLG)*SL+FLOAT(NSG)*SS+FLOAT(NLG+NSG)*SP	DH 92
		NLT=IFIX((ST+SP)/DT)	DH 93
		IF (NLT.LE.0) NLT=1	DH 94
95		DTN=(ST+SP)/FLOAT(NLT)	DH 95
		TI=DTN/DT	DH 96
		SL=SL*TI	DH 97
		SS=SS*TI	DH 98
		SP=SP*TI	DH 99
100	C	POSITION PEN AT START OF LINE	DH 100
		SI=0.	DH 101
		CALL CALPLT (XI(1),YI(1),3)	DH 102
		JSTART=1	DH 103
	C	PLOT NL LONG DASHES	DH 104
105	10	DO 11 I=1,NLG	DH 105
		SI=SI+SL	DH 106
		IF (SI+EPS.GE.ST) SI=ST	DH 107
		CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)	DH 108
		CALL CALPLT (XX,YY,2)	DH 109
110		SI=SI+SP	DH 110
		IF (SI.GE.ST) GO TO 14	DH 111
		CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)	DH 112
		CALL CALPLT (XX,YY,3)	DH 113
	11	CONTINUE	DH 114
115	C	PLOT NS SHORT DASHES	DH 115
		IF (NSG.EQ.0) GO TO 10	DH 116
		DO 12 I=1,NSG	DH 117
		SI=SI+SS	DH 118
		IF (SI+EPS.GE.ST) SI=ST	DH 119
120		CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)	DH 120

CARD NO.

121		CALL CALPLT (XX,YY,2)	DH 121
		SI=SI+SP	DH 122
		IF (SI.GE.ST) GO TO 14	DH 123
		CALL LINEAR (SI,XX,YY,NP,S,XI,YI,JSTART)	DH 124
125		CALL CALPLT (XX,YY,3)	DH 125
	12	CONTINUE	DH 126
	C	CONTINUE PLOTTING DASH LINE	DH 127
		GO TO 10	DH 128
	C	PLOT CONTINUOUS LINE	DH 129
130	13	XI(NP+1)=0.0	DH 130
		YI(NP+1)=0.0	DH 131
		XI(NP+2)=YI(NP+2)=1.0	DH 132
		CALL LINE (XI,YI,NP,1,0,0,0.)	DH 133
	C		DH 134
135	C	RETURN TO START OF CURVE	DH 135
	C		DH 136
	14	CALL CALPLT (XI(1),YI(1),3)	DH 137
	C		DH 138
	C	RETURN TO CALLING PROGRAM	DH 139
140	C		DH 140
		RETURN	DH 141
	C		DH 142
	15	FORMAT (1H1//5X,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT	DH 143
		1,13X,1HX,13X,1HY/)	DH 144
145	16	FORMAT (5X,3F15.4)	DH 145
		END	DH 146-

LISTING OF DECK: LINEAR

PAGE 1

CARD NO.

1		SUBROUTINE LINEAR (TINT,XINT,YINT,N,T,X,Y,JSTART)	LR	1
	C		LR	2
	C	LINEAR INTERPOLATION METHOD	LR	3
	C		LR	4
5	C	CODED BY -- HARRY L. MORGAN	LR	5
	C	NASA/LARC/TAD/AA9	LR	6
		1983	LR	7
		DIMENSION T(1), X(1), Y(1)	LR	8
		J=1	LR	9
		IF (TINT.LE.T(1)) GO TO 3	LR	10
10		J=N	LR	11
		IF (TINT.GE.T(N)) GO TO 3	LR	12
		JEND=JSTART	LR	13
		DO 1 I=JSTART,N	LR	14
		J=I	LR	15
15		IF (T(J)-TINT) 1,3,2	LR	16
	1	JEND=J	LR	17
	2	DT=(TINT-T(J-1))/(T(J)-T(J-1))	LR	18
		XINT=Y(J-1)+(X(J)-X(J-1))*DT	LR	19
		YINT=Y(J-1)+(Y(J)-Y(J-1))*DT	LR	20
20		JSTART=JEND	LR	21
		RETURN	LR	22
	3	YINT=Y(J)	LR	23
		XINT=X(J)	LR	24
		JSTART=J	LR	25
25		RETURN	LR	26-
		END		

CARD NO.

1	SUBROUTINE LSQPLT (T,X,Y,N,NSYM,ISIZE,IOP,IRUN,TENSION,ILSQ,NPOL,D	LS	1
	1F)	LS	2
	C	LS	3
	C ROUTINE TO FAIR DATA WITH LEAST SQUARES CURVE FIT	LS	4
5	C	LS	5
	C CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAR 1983	LS	6
	C	LS	7
	C PARAMETER DEFINITION	LS	8
	C T - ARRAY CONTAINING INDEPENDENT VARIABLE	LS	9
10	C X - ARRAY CONTAINING X VARIABLE	LS	10
	C Y - ARRAY CONTAINING Y VARIABLE	LS	11
	C N - NUMBER OF POINTS TO BE PLOTTED	LS	12
	C NSYM - LINE AND SYMBOL NUMBER	LS	13
	C ISIZE - SYMBOL SIZE	LS	14
15	C IOP - PLOTTING OPTION IOP=0 PLOT SYMBOLS ONLY	LS	15
	C IOP=1 PLOT SYMBOLS AND CURVE	LS	16
	C IRUN - RUN NUMBER	LS	17
	C TENSION - SPLINE TENSION	LS	18
	C ILSQ - LEAST SQUARES CURVE FIT OPTION	LS	19
20	C ILSQ=0 LEAST SQUARES POLYNOMIAL CURVE FIT	LS	20
	C ILSQ=1 LEAST SQUARES CUBIC SPLINE FIT	LS	21
	C NPOL - ORDER OF LEAST SQUARES POLYNOMIAL CURVE FIT	LS	22
	C DF - STANDARD DEVIATION FOR LEAST-SQUARES CUBIC-SPLINE FIT	LS	23
	C DIMENSION T, X, AND Y IN CALLING PROGRAM	LS	24
25	C DIMENSION T(1), X(1), Y(1)	LS	25
	C WORK ARRAYS	LS	26
	C DIMENSION XNEW(50), YNEW(50)	LS	27
	C DIMENSION COEF(50,4)	LS	28
	C COMMON /WORK/ WK(360),SDV(50),CX(11),CY(11)	LS	29
30	C	LS	30
	C ROUTINE CONSTANTS	LS	31
	C	LS	32
	C NMAX=50	LS	33
	C WT=1.0	LS	34
35	C IPRINT=0	LS	35
	C	LS	36
	C SELECT PLOTTING OPTION	LS	37
	C	LS	38
	C IF (IOP.EQ.0.OR.N.LT.3) GO TO 1	LS	39
40	C GO TO 3	LS	40

CARD NO.

41	C		LS 41
	C	PLOT SYMBOLS ONLY	LS 42
	C		LS 43
	1	DO 2 I=1,N	LS 44
45	2	CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)	LS 45
		RETURN	LS 46
	C		LS 47
	C	CHECK TO SEE IF T IS STRICTLY INCREASING	LS 48
	C		LS 49
50	3	DO 4 I=2,N	LS 50
		IF (T(I).LT.T(I-1)) GO TO 5	LS 51
	4	CONTINUE	LS 52
		GO TO 6	LS 53
	5	PRINT 20, IRUN	LS 54
55		PRINT 21, (T(I),X(I),Y(I),I=1,N)	LS 55
		GO TO 1	LS 56
	C		LS 57
	C	PLOT DATA POINTS WITH SYMBOLS	LS 58
	C		LS 59
60	6	DO 7 I=1,N	LS 60
		SDV(I)=WT	LS 61
		IF (ILSQ.EQ.1) SDV(I)=DF	LS 62
		CALL PNTPLT (X(I),Y(I),NSYM,ISIZE)	LS 63
	7	CONTINUE	LS 64
65	C		LS 65
	C	COMPUTE LEAST SQUARES CURVE	LS 66
	C		LS 67
		IF (ILSQ.GT.0) GO TO 10	LS 68
	C	COMPUTE COEFFICIENTS OF LEAST SQUARES POLYNOMIAL	LS 69
70		MPOL=NPOL	LS 70
		IF (MPOL.GT.N-1) MPOL=N-1	LS 71
		IF (MPOL.LE.0) MPOL=1	LS 72
		IF (MPOL.GT.10) MPOL=10	LS 73
		M1=MPOL+1	LS 74
75		CALL LSO (T,X,SDV,N,MPOL,CX)	LS 75
		CALL LSO (T,Y,SDV,N,MPOL,CY)	LS 76
	C	COMPUTE NEW X AND Y VALUES AND ERROR SUMMATIONS	LS 77
		ERRX=ERRY=0.0	LS 78
		DO 9 I=1,N	LS 79
80		YNEW(I)=CX(1)	LS 80

CARD NO.

81		YNEW(I)=CY(1)	LS 81
		DELTA=1.0	LS 82
		DO 8 J=2,M1	LS 83
		DELTA=DELTA*T(I)	LS 84
85		XNEW(I)=XNEW(I)+DELTA*CX(J)	LS 85
	8	YNEW(I)=YNEW(I)+DELTA*CY(J)	LS 86
		ERRX=ERRX+(X(I)-XNEW(I))**2	LS 87
		ERRY=ERRY+(Y(I)-YNEW(I))**2	LS 88
	9	CONTINUE	LS 89
90		GO TO 17	LS 90
	C	COMPUTE COEFFICIENTS OF LEAST SQUARES CUBIC SPLINE	LS 91
	C	FIT X	LS 92
	10	K=-1	LS 93
		SM=FLOAT(N)	LS 94
95		CALL CSDS (NMAX,N,T,X,SDV,SM,K,COEF,WK,IERR)	LS 95
		IF (IERR.NE.0) GO TO 19	LS 96
	C	COMPUTE NEW X AND ERROR SUMMATION	LS 97
		ERRX=0.	LS 98
		DO 13 I=1,N	LS 99
100		IF (I.EQ.N) GO TO 11	LS 100
		XNEW(I)=COEF(I,1)	LS 101
		GO TO 12	LS 102
	11	K=I-1	LS 103
		DELTA=T(I)-T(K)	LS 104
105		XNEW(I)=((COEF(K,4)*DELTA+COEF(K,3))*DELTA+COEF(K,2))*DELTA+COEF(K,1)	LS 105
	12	ERRX=ERRX+(X(I)-XNEW(I))**2	LS 106
	13	CONTINUE	LS 107
	C	FIT Y	LS 108
110		K=-1	LS 109
		SM=FLOAT(N)	LS 110
		CALL CSDS (NMAX,N,T,Y,SDV,SM,K,COEF,WK,IERR)	LS 111
		IF (IERR.NE.0) GO TO 19	LS 112
	C	COMPUTE NEW Y AND ERROR SUMMATION	LS 113
115		ERRY=0.	LS 114
		DO 16 I=1,N	LS 115
		IF (I.EQ.N) GO TO 14	LS 116
		YNEW(I)=COEF(I,1)	LS 117
		GO TO 15	LS 118
120	14	K=I-1	LS 119
			LS 120

CARD NO.

121	DELTA=T(I)-T(K)	LS 121
	YNEW(I)=((COEF(K,4)*DELTA+COEF(K,3))*DELTA+COEF(K,2))*DELTA+COEF(K	LS 122
	1,1)	LS 123
15	ERRY=ERRY+(Y(I)-YNEW(I))*2	LS 124
125	16 CONTINUE	LS 125
	C	LS 126
	C PLOT DASHED LINE THRU NEW X AND Y VALUES	LS 127
	C	LS 128
17	TEN=TENSION	LS 129
130	IF (ILSQ.EQ.1) TEN=0.0	LS 130
	CALL DASHLN (T,XNEW,YNEW,N,NSYM,IOP,ISIZE,IRUN,TEN)	LS 131
	C	LS 132
	C PRINT RESULTS OF LEAST SQUARES CURVE FIT	LS 133
	C	LS 134
135	IF (IPRINT.EQ.0) RETURN	LS 135
	IF (ILSQ.EQ.0) PRINT 22, IRUN,NSYM,MPOL	LS 136
	IF (ILSQ.EQ.1) PRINT 23, IRUN,NSYM,DF	LS 137
	DO 18 I=1,N	LS 138
	DELTA=X(I)-XNEW(I)	LS 139
140	SM=Y(I)-YNEW(I)	LS 140
	PRINT 24, I,T(I),X(I),XNEW(I),DELTA,Y(I),YNEW(I),SM	LS 141
	18 CONTINUE	LS 142
	PRINT 25, ERY,ERRY	LS 143
	RETURN	LS 144
145	C	LS 145
	C PRINT ERROR MESSAGE	LS 146
	C	LS 147
	19 PRINT 26, IERR,IRUN,NSYM	LS 148
	GO TO 1	LS 149
150	C	LS 150
	20 FORMAT (1H1//5X,36HT IS NOT STRICTLY INCREASING FOR RUN,I7/16X,1HT	LS 151
	1,13X,1HX,13X,1HY//	LS 152
	21 FORMAT (5X,3F15.4)	LS 153
	22 FORMAT (/5X,55HRESULTS OF LEAST SQUARES POLYNOMIAL CURVE FIT FOR	LS 154
155	1RUN =,I4,3X,6HNSYM =,I2,3X,6HNPOL =,I3//4X,1HI,6X,1HT,9X,1HX,9X,4H	LS 155
	2XNEW,4X,6HX-XNEW,6X,1HY,9X,4HYNEW,4X,6HY-YNEW)	LS 156
	23 FORMAT (/5X,57HRESULTS OF LEAST SQUARES CUBIC SPLINE CURVE FIT FO	LS 157
	1R RUN =,I4,3X,6HNSYM =,I2,3X,4HDF =,F10.5//4X,1HI,6X,1HT,9X,1HX,9X	LS 158
	2,4HXNEW,4X,6HX-XNEW,6X,1HY,9X,4HYNEW,4X,6HY-YNEW)	LS 159
160	24 FORMAT (I5,7F10.4)	LS 160

CARD NO.

161	25	FORMAT (/10X,12HERROR IN X =,E15.6,10X,12HERROR IN Y =,E15.6)	LS 161
	26	FORMAT (/5X,13HERROR NUMBER ,I3,42H OCCURRED IN CSOS CALL IN LSQPL	LS 162
		1T FOR RUN =,I4,11H AND NSYM =,I3)	LS 163
		END	LS 164-

CARD NO.

1		SUBROUTINE LSQ (X,Y,W,NP,N,C)	LQ	1
	C		LQ	2
	C	LEAST SQUARES WEIGHTED POLYNOMIAL CURVE FIT ROUTINE	LQ	3
	C		LQ	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	LQ	5
	C		LQ	6
	C	PARAMETER DEFINITION	LQ	7
	C	X - ARRAY CONTAINING INDEPENDENT VARIABLE	LQ	8
	C	Y - ARRAY CONTAINING DEPENDENT VARIABLE	LQ	9
10	C	W - ARRAY CONTAINING WEIGHTING VALUES	LQ	10
	C	NP - NUMBER OF POINTS	LQ	11
	C	N - ORDER OF POLYNOMIAL	LQ	12
	C	C - COEFFICIENTS OF POLYNOMIAL	LQ	13
	C		LQ	14
15	C	NEW Y VALUE = C(1)+C(2)*X+C(3)*X**2+...+C(N+1)*X**N	LQ	15
	C		LQ	16
	C	DIMENSION X(NP), Y(NP), W(NP), AND C(N+1) IN CALLING PROGRAM	LQ	17
	C	DIMENSION X(1), Y(1), W(1), C(1)	LQ	18
	C	DIMENSION A(N+1,N+2) N=10 FOR THIS VERSION	LQ	19
20	C	COMMON /WORK/ A(11,12)	LQ	20
	C		LQ	21
	C	COMPUTE LEAST SQUARES MATRIX	LQ	22
	C		LQ	23
		N1=N+1	LQ	24
25		N2=N+2	LQ	25
		DO 1 I=1,N1	LQ	26
		DO 1 J=1,N2	LQ	27
	1	A(I,J)=0.	LQ	28
		DO 3 K=1,NP	LQ	29
30		T1=1.	LQ	30
		DO 3 J=1,N1	LQ	31
		T2=T1	LQ	32
		DO 2 I=1,N1	LQ	33
		A(J,I)=A(J,I)+T2*W(K)	LQ	34
35	2	T2=T2*X(K)	LQ	35
		A(J,N2)=A(J,N2)-Y(K)*T1*W(K)	LQ	36
	3	T1=T1*X(K)	LQ	37
	C		LQ	38
	C	SOLVE FOR COEFFICIENTS OF LEAST SQUARES POLYNOMIAL	LQ	39
40	C		LQ	40

LISTING OF DECK: LSQ

PAGE 2

CARD NO.

41		DO 4 K=1,N	LQ 41
		DO 4 J=K,N	LQ 42
		T1=A(J+1,K)/A(K,K)	LQ 43
		DO 4 I=K,N2	LQ 44
45	4	A(J+1,I)=A(J+1,I)-A(K,I)*T1	LQ 45
		C(N1)=-A(N1,N2)/A(N1,N1)	LQ 46
		DO 5 I=2,N1	LQ 47
		K=N2-I	LQ 48
		C(K)=-A(K,N2)/A(K,K)	LQ 49
50		L=K+1	LQ 50
		DO 5 J=L,N1	LQ 51
	5	C(K)=C(K)-C(J)*A(K,J)/A(K,K)	LQ 52
		RETURN	LQ 53
		END	LQ 54-

CARD NO.

1		SUBROUTINE CSDS (MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)	CS	1
	C*****		CS	2
	C*		* CS	3
	C*	PURPOSE:	* CS	4
5	C*	SUBROUTINE CSDS FITS A SMOOTH CUBIC SPLINE TO A	* CS	5
	C*	UNIVARIATE FUNCTION. DATA MAY BE UNEQUALLY SPACED.	* CS	6
	C*		* CS	7
	C*	USE:	* CS	8
	C*	CALL CSDS(MAX,IX,X,F,DF,S,IPT,COEF,WK,IERR)	* CS	9
10	C*		* CS	10
	C*	MAX INPUT INTEGER SPECIFYING THE MAXIMUM NUMBER OF DATA	* CS	11
	C*	POINTS FOR THE INDEPENDENT VARIABLE.	* CS	12
	C*		* CS	13
	C*	IX INPUT INTEGER SPECIFYING THE ACTUAL NUMBER OF DATA	* CS	14
15	C*	POINTS FOR THE INDEPENDENT VARIABLE. IX≤MAX.	* CS	15
	C*		* CS	16
	C*	X ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST	* CS	17
	C*	IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS,	* CS	18
	C*	X(I) MUST CONTAIN THE VALUE OF THE INDEPENDENT	* CS	19
20	C*	VARIABLE AT POINT I.	* CS	20
	C*		* CS	21
	C*	F ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST	* CS	22
	C*	IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS,	* CS	23
	C*	F(I) MUST CONTAIN THE VALUE OF THE FUNCTION AT	* CS	24
25	C*	POINT X(I).	* CS	25
	C*		* CS	26
	C*	DF ONE-DIMENSIONAL INPUT ARRAY DIMENSIONED AT LEAST	* CS	27
	C*	IX IN THE CALLING PROGRAM. UPON ENTRY TO CSDS,	* CS	28
	C*	DF(I) MUST CONTAIN AN ESTIMATE OF THE STANDARD	* CS	29
30	C*	DEVIATION OF F(I).	* CS	30
	C*		* CS	31
	C*	S A NON-NEGATIVE INPUT PARAMETER WHICH CONTROLS THE	* CS	32
	C*	EXTENT OF SMOOTHING. S SHOULD BE IN THE RANGE	* CS	33
	C*	(IX-(2*IX)**.5)<S<(IX+(2*IX)**.5).	* CS	34
35	C*		* CS	35
	C*	IPT INPUT INITIALIZATION PARAMETER. THE USER MUST	* CS	36
	C*	SPECIFY IPT=-1 WHENEVER A NEW X ARRAY IS	* CS	37
	C*	INPUT. THE ROUTINE WILL ALSO CHECK TO INSURE THAT	* CS	38
	C*	THE X ARRAY IS IN STRICTLY INCREASING ORDER.	* CS	39
40	C*		* CS	40

CARD NO.

41	C*	COEF	A TWO-DIMENSIONAL OUTPUT ARRAY DIMENSIONED (MAX,4)	* CS	41	
	C*		IN THE CALLING PROGRAM. UPON RETURN, COEF(I,J)	* CS	42	
	C*		CONTAINS THE J-TH COEFFICIENT OF THE SPLINE FOR	* CS	43	
	C*		THE INTERVAL BEGINNING AT POINT X(I). THE	* CS	44	
45	C*		FUNCTIONAL VALUE OF THE SPLINE AT ABSCISSA X1,	* CS	45	
	C*		WHERE $X(I) < X1 < X(I+1)$, IS GIVEN BY:	* CS	46	
	C*		$F(X1) = ((COEF(I,4)*H + COEF(I,3))*H + COEF(I,2))*H$	* CS	47	
	C*		+COEF(I,1)	* CS	48	
	C*		WHERE $H = X1 - X(I)$	* CS	49	
50	C*			* CS	50	
	C*	WK	A ONE-DIMENSIONAL WORK AREA ARRAY DIMENSIONED AT	* CS	51	
	C*		LEAST (7*IX+9) IN THE CALLING PROGRAM.	* CS	52	
	C*			* CS	53	
	C*	IERR	OUTPUT ERROR PARAMETER:	* CS	54	
55	C*		=0 NORMAL RETURN. NO ERROR DETECTED.	* CS	55	
	C*		=J THE J-TH ELEMENT OF THE X ARRAY IS NOT IN	* CS	56	
	C*		STRICTLY INCREASING ORDER.	* CS	57	
	C*		=-1 THERE ARE LESS THAN FOUR VALUES IN THE X ARRAY.	* CS	58	
	C*			* CS	59	
60	C*		UPON RETURN FROM CSDS, THIS PARAMETER SHOULD BE	* CS	60	
	C*		TESTED IN THE CALLING PROGRAM.	* CS	61	
	C*			* CS	62	
	C*			* CS	63	
	C*			* CS	64	
65	C*	REQUIRED ROUTINES	-NONE	* CS	65	
	C*			* CS	66	
	C*	SOURCE	IMSL ROUTINE ICSSMU MODIFIED BY	* CS	67	
	C*		COMPUTER SCIENCES CORPORATION	* CS	68	
	C*			* CS	69	
70	C*	LANGUAGE	-FORTRAN	* CS	70	
	C*			* CS	71	
	C*	DATE RELEASED	SEPTEMBER 5, 1973	* CS	72	
	C*			* CS	73	
	C*	LATEST REVISION	MARCH 1975	* CS	74	
75	C*	*****			* CS	75
	C				CS	76
	C				CS	77
	C	DIMENSION X(1), F(1), DF(1), COEF(MAX,4), WK(1)			CS	78
	C				CS	79
80	C	SET UP WORKING AREAS			CS	80

CARD NO.

81	C	IERR=0	CS 81
		IF (IPT.NE.-1) GO TO 4	CS 82
		IPT=0	CS 83
85		IF (IX.LT.4) GO TO 1	CS 84
		GO TO 2	CS 85
	1	IERR=-1	CS 86
		RETURN	CS 87
	2	IX1=IX-1	CS 88
90		DO 3 I=1,IX1	CS 89
		IF (X(I+1)-X(I)).GT.0) GO TO 3	CS 90
		IERR=I+1	CS 91
		RETURN	CS 92
	3	CONTINUE	CS 93
95		NP1=IX+1	CS 94
		IB1=NP1	CS 95
		IB2=IB1+NP1	CS 96
		IB3=IB2+NP1+1	CS 97
		IB4=IB3+NP1	CS 98
100		IB5=IB4+NP1	CS 99
		IB6=IB5+NP1+1	CS 100
		WK(1)=0.	CS 101
		WK(2)=0.	CS 102
		WK(1B2)=0.	CS 103
105		WK(1B3)=0.	CS 104
		IJK2=IB2+NP1	CS 105
		WK(IJK2)=0.	CS 106
		IJK5=IB5+1	CS 107
		WK(IJK5)=0.	CS 108
110		IJK5=IB5+2	CS 109
		WK(IJK5)=0.	CS 110
		WK(1B6)=0.	CS 111
		IJK5=IB5+NP1	CS 112
		WK(IJK5)=0.	CS 113
115	4	CONTINUE	CS 114
		P=0.	CS 115
		H=X(2)-X(1)	CS 116
		F2=-S	CS 117
		FF=(F(2)-F(1))/H	CS 118
120		IF (IX.LT.3) GO TO 10	CS 119
			CS 120

CARD NO.

121		DO 5 I=3, IX	CS 121
		G=H	CS 122
		H=X(I)-X(I-1)	CS 123
		E=FF	CS 124
125		FF=(F(I)-F(I-1))/H	CS 125
		COFF(I-1,1)=FF-E	CS 126
		IJK3=IB3+I	CS 127
		WK(IJK3)=(G+H)*.666666666666667	CS 128
		IJK4=IB4+I	CS 129
130		WK(IJK4)=H/3.	CS 130
		IJK2=IB2+I	CS 131
		WK(IJK2)=DF(I-2)/G	CS 132
		WK(I)=DF(I)/H	CS 133
		IJK1=IB1+I	CS 134
135		WK(IJK1)=-DF(I-1)/G-DF(I-1)/H	CS 135
	5	CONTINUE	CS 136
		DO 6 I=3, IX	CS 137
		IJK1=IB1+I	CS 138
		IJK2=IB2+I	CS 139
140		COEF(I-1,2)=WK(I)*WK(I)+WK(IJK1)*WK(IJK1)+WK(IJK2)*WK(IJK2)	CS 140
		COEF(I-1,3)=WK(I)*WK(IJK1+1)+WK(IJK1)*WK(IJK2+1)	CS 141
		COEF(I-1,4)=WK(I)*WK(IJK2+2)	CS 142
	6	CONTINUE	CS 143
	C		CS 144
145	C	NEXT ITERATION	CS 145
	C		CS 146
	7	IF (IX.LT.3) GO TO 10	CS 147
		DO 8 I=3, IX	CS 148
		IJK1=IB1+I-1	CS 149
150		IJK0=I-1	CS 150
		WK(IJK1)=FF*WK(IJK0)	CS 151
		IJK2=IB2+I-2	CS 152
		IJK0=I-2	CS 153
		WK(IJK2)=G*WK(IJK0)	CS 154
155		IJK0=I	CS 155
		IJK3=IB3+I	CS 156
		WK(IJK0)=1./((P*COEF(I-1,2)+WK(IJK3)-FF*WK(IJK1)-G*WK(IJK2))	CS 157
		IJK5=IB5+I	CS 158
		IJKN=IJK5-1	CS 159
160		IJK0=IJKN-1	CS 160

CARD NO.

161		WK(IJK5)=COEF(I-1,1)-WK(IJK1)*WK(IJKN)-WK(IJK2)*WK(IJK0)	CS 161
		IJK4=IB4+I	CS 162
		FF=P*COEF(I-1,3)+WK(IJK4)-H*WK(IJK1)	CS 163
		G=H	CS 164
165		H=COEF(I-1,4)*P	CS 165
	8	CONTINUE	CS 166
		DO 9 I=3,IX	CS 167
		J=IX-I+3	CS 168
		IJK5=IB5+J	CS 169
170		IJK6=IJK5+1	CS 170
		IJK7=IJK6+1	CS 171
		IJK1=IB1+J	CS 172
		IJK2=IB2+J	CS 173
		WK(IJK5)=WK(J)*WK(IJK5)-WK(IJK1)*WK(IJK6)-WK(IJK2)*WK(IJK7)	CS 174
175	9	CONTINUE	CS 175
	10	E=0	CS 176
		H=0	CS 177
	C		CS 178
	C	COMPUTE U AND ACCUMULATE E	CS 179
180	C		CS 180
		DO 11 I=2,IX	CS 181
		G=H	CS 182
		IJK5=IB5+I	CS 183
		H=(WK(IJK5+1)-WK(IJK5))/(X(I)-X(I-1))	CS 184
185		IJK6=IB6+I	CS 185
		WK(IJK6)=(H-G)*DF(I-1)*DF(I-1)	CS 186
		E=E+WK(IJK6)*(H-G)	CS 187
	11	CONTINUE	CS 188
		G=-H*DF(IX)*DF(IX)	CS 189
190		IJK6=IB6+NP1	CS 190
		WK(IJK6)=G	CS 191
		E=E-G*H	CS 192
		G=F2	CS 193
		F2=E*P*P	CS 194
195		IF (F2.GE.S.OR.F2.LE.G) GO TO 14	CS 195
		FF=0.	CS 196
		IJK6=IB6+2	CS 197
		H=(WK(IJK6+1)-WK(IJK6))/(X(2)-X(1))	CS 198
		IF (IX.LT.3) GO TO 13	CS 199
200		DO 12 I=3,IX	CS 200

CARD NO.

201		G=H	CS 201
		IJK6=IB6+I	CS 202
		H=(WK(IJK6+1)-WK(IJK6))/(X(I)-X(I-1))	CS 203
		IJK1=IB1+I-1	CS 204
205		IJK2=IB2+I-2	CS 205
		G=H-G-WK(IJK1)*WK(I-1)-WK(IJK2)*WK(I-2)	CS 206
		FF=FF+G*WK(I)*G	CS 207
		WK(I)=G	CS 208
	12	CONTINUE	CS 209
210	13	H=E-P*FF	CS 210
		IF (H.LE.0) GO TO 14	CS 211
	C		CS 212
	C	UPDATE THE LAGRANGE MULTIPLIER P	CS 213
	C	FOR THE NEXT ITERATION	CS 214
215	C		CS 215
		P=P+(S-F2)/(((SQRT(S/E)+P)*H)	CS 216
		GO TO 7	CS 217
	C		CS 218
	C	IF E LESS THAN OR EQUAL TO S,	CS 219
220	C	COMPUTE THE COEFFICIENTS AND RETURN.	CS 220
	C		CS 221
	14	DO 15 I=2,NP1	CS 222
		IJK6=IB6+I	CS 223
		COEF(I-1,1)=F(I-1)-P*WK(IJK6)	CS 224
225		IJK5=IB5+I	CS 225
		COEF(I-1,3)=WK(IJK5)	CS 226
	15	CONTINUE	CS 227
		DO 16 I=2,IX	CS 228
		H=X(I)-X(I-1)	CS 229
230		COEF(I-1,4)=(COEF(I,3)-COEF(I-1,3))/(3.*H)	CS 230
		COEF(I-1,2)=(COEF(I,1)-COEF(I-1,1))/H-(H*COEF(I-1,4)+COEF(I-1,3))*	CS 231
	1H		CS 232
	16	CONTINUE	CS 233
		RETURN	CS 234
235		END	CS 235-

LISTING OF DECK: AXISLB

PAGE 1

CARD NO.

1		SUBROUTINE AXISLB (IAXIS,XORG,YORG,ORG,SCALE,DIST,HT,NDIG)	XL	1
	C		XL	2
	C	ROUTINE TO DRAW SCALE VALUES ON AXIS	XL	3
	C		XL	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	XL	5
	C		XL	6
	C	PARAMETER DEFINITION	XL	7
	C	IAXIS = 1 - DRAW X-AXIS SCALES	XL	8
	C	IAXIS = 2 - DRAW Y-AXIS SCALES	XL	9
10	C	XORG,YORG - X,Y COORDINATES OF BEGINNING OF AXIS	XL	10
	C	ORG - SCALE VALUE AT BEGINNING OF AXIS	XL	11
	C	SCALE - SCALE VALUE PER INCH	XL	12
	C	DIST - LENGTH OF AXIS	XL	13
	C	HT - HEIGHT OF VALUES DRAWN ON AXIS	XL	14
15	C	NDIG - NUMBER OF SIGNIFICANT FIGURES TO RIGHT OF DECIMAL POINT	XL	15
	C	NDIG=-1 WILL DROP DECIMAL POINT	XL	16
	C		XL	17
	C	COMPUTE ROUTINE CONSTANTS	XL	18
		WADJ=0.0	XL	19
20		IF (HT.LE.0.1) WADJ=0.01	XL	20
		CONS=6.*HT/7.+WADJ	XL	21
		CONS2=2.*HT/7.	XL	22
		ADJ=0.4*HT	XL	23
		IF (ADJ.LT.0.1) ADJ=0.1	XL	24
25		NL=IFIX(DIST+0.0001)+1	XL	25
		XO=XORG	XL	26
		YO=YORG	XL	27
		VALUE=ORG	XL	28
	C	POSITION PEN AT BEGINNING OF AXIS	XL	29
30		CALL CALPLT (XORG,YORG,3)	XL	30
	C	DRAW AXIS VALUE AT EACH INCH MARK ALONG AXIS	XL	31
		DO P I=1,NL	XL	32
		VALUE=SCALE*FLOAT(I-1)+ORG	XL	33
	C	COMPUTE NUMBER OF SIGNIFICANT FIGURES IN VALUE	XL	34
35		NDG=NDIG	XL	35
		V=ABS(VALUE)	XL	36
		IF (V-1.0E-5) 1,1,2	XL	37
	1	NDG=-1	XL	38
		NSIG=1	XL	39
40		V=0.0	XL	40

CARD NO.

41		VALUE=0.0	XL 41
		GO TO 5	XL 42
	2	NSIG=NDG+1	XL 43
	3	S=V+SIGN(1.E-7,V)	XL 44
45		N=IFIX(S)	XL 45
		IF (N) 5,5,4	XL 46
	4	NSIG=NSIG+1	XL 47
		V=0.10*V	XL 48
		GO TO 3	XL 49
50	5	IF (VALUE.LT.0.0) NSIG=NSIG+1	XL 50
		S=CONS*FLOAT(NSIG)-CONS2	XL 51
	C	POSITION AND DRAW AXIS VALUE	XL 52
		IF (IAXIS.EQ.2) GO TO 6	XL 53
		XT=XO-0.5*S	XL 54
55		YT=Y0-HT-ADJ	XL 55
		XO=XO+1.	XL 56
		GO TO 7	XL 57
	6	XT=XO-S-ADJ	XL 58
		YT=Y0-0.5*HT	XL 59
60		Y0=Y0+1.	XL 60
	7	CALL NUMBER (XT,YT,HT,VALUE,0.,NDG)	XL 61
	C	ADVANCE TO NEXT INCH MARK	XL 62
	8	CONTINUE	XL 63
	C	RETURN TO BEGINNING OF AXIS	XL 64
65		CALL CALPLT (XORG,YORG,3)	XL 65
	C	RETURN TO CALLING PROGRAM	XL 66
		RETURN	XL 67
		END	XL 68-

CARD NO.

1		SUBROUTINE COEFSY (XO,YO,HT,ISYM)	CY	1
	C		CY	2
	C	ROUTINE TO DRAW AERODYNAMIC COEFFICIENTS	CY	3
	C		CY	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	CY	5
	C		CY	6
	C	PARAMETER DEFINITION	CY	7
	C	XO - X COORDINATE OF LEFT-HAND EDGE OF SYMBOL	CY	8
	C	YO - Y COORDINATE OF CENTERLINE OF MAIN LETTER OF SYMBOL	CY	9
10	C	HT - SYMBOL HEIGHT	CY	10
	C	ISYM = 1 - LIFT WIDTH = 2.0*HT	CY	11
	C	ISYM = 2 - DRAG WIDTH = 2.1*HT	CY	12
	C	ISYM = 3 - PITCHING MOMENT WIDTH = 2.2*HT	CY	13
	C	ISYM = 4 - ROLLING MOMENT WIDTH = 1.7*HT	CY	14
15	C	ISYM = 5 - YAWING MOMENT WIDTH = 1.9*HT	CY	15
	C	ISYM = 6 - SIDE FORCE WIDTH = 2.1*HT	CY	16
	C	ISYM = 7 - LIFT/DRAG WIDTH = 2.6*HT	CY	17
	C	ISYM = 8 - ALPHA, DEG WIDTH = 3.5*HT	CY	18
	C	ISYM = 9 - BETA, DEG WIDTH = 3.4*HT	CY	19
20	C		CY	20
	C	INITIALIZE CHARACTER SETS	CY	21
		CALL CHARST1	CY	22
		CALL CHARST7	CY	23
		CALL CHARST2	CY	24
25	C	POSITION PEN AT START OF SYMBOL	CY	25
		CALL CALPLT (XO,YO,3)	CY	26
		Y1=YO-0.5*HT	CY	27
		IF (ISYM.EQ.7) GO TO 1	CY	28
		IF (ISYM.EQ.8.OR.ISYM.EQ.9) GO TO 2	CY	29
30		IF (ISYM.GT.9) GO TO 3	CY	30
	C	DRAW LETTER C	CY	31
		CALL CHARACT (XO,Y1,HT,2H)C,0.,2,0.0)	CY	32
		CALL CHARWH (WD,X1,Y1,HT,2H)C,2,0.0)	CY	33
		X1=XO+1.15*WD	CY	34
35		Y1=YO-HT	CY	35
	C	DRAW SUBSCRIPT OF C	CY	36
		IF (ISYM.EQ.1) CALL CHARACT (X1,Y1,HT,2H)L,0.,2,0.0)	CY	37
		IF (ISYM.EQ.2) CALL CHARACT (X1,Y1,HT,2H)D,0.,2,0.0)	CY	38
		IF (ISYM.EQ.3) CALL CHARACT (X1,Y1,HT,2H)M,0.,2,0.0)	CY	39
40		IF (ISYM.EQ.4) CALL LAM (X1,Y1,HT)	CY	40

CARD NO.

41		IF (ISYM.EQ.5) CALL CHARACT (X1,Y1,HT,2H(N,0.,2,0.0)	CY	41
		IF (ISYM.EQ.6) CALL CHARACT (X1,Y1,HT,2H)Y,0.,2,0.0)	CY	42
		GO TO 4	CY	43
	C	DRAW L/D	CY	44
45	1	CALL CHARACT (X0,Y1,HT,4H)L/D,0.,4,0.25)	CY	45
		GO TO 4	CY	46
	C	DRAW ALPHA,DEG OR BETA,DEG	CY	47
	2	IF (ISYM.EQ.8) CALL CHARACT (X0,Y1,HT,11H\$7(A\$2(,DEG,0.,11,0.20)	CY	48
		IF (ISYM.EQ.9) CALL CHARACT (X0,Y1,HT,11H\$7(B\$2(,DEG,0.,11,0.20)	CY	49
50		GO TO 4	CY	50
	C		CY	51
	C	ADD ADDITIONAL SYMBOLS HERE AS DESIRED	CY	52
	3	CONTINUE	CY	53
	C		CY	54
55	C		CY	55
	C	RETURN PEN TO START OF SYMBOL	CY	56
	4	CALL CALPLT (X0,Y0,3)	CY	57
		CALL CHNGSET (1)	CY	58
	C	RETURN TO CALLING PROGRAM	CY	59
60		RETURN	CY	60
		END	CY	61-

LISTING OF DECK: LAM

PAGE 1

CARD NO.

1		SUBROUTINE LAM (XO,YO,HT)	LM	1
	C		LM	2
	C	ROUTINE TO DRAW ROLLING MOMENT SUBSCRIPT	LM	3
	C		LM	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	LM	5
	C		LM	6
	C	PARAMETER DEFINITION	LM	7
	C	XO,YO - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF SYMBOL	LM	8
	C	HT - HEIGHT OF SYMBOL	LM	9
10	C		LM	10
		DIMENSION XM(17), YM(17)	LM	11
		DATA XM/3.,13.,0.,0.,4.,10.,13.,14.,13.,12.2,10.,4.,1.,1.,14.4,3.6	LM	12
		1,3./	LM	13
		DATA YM/43.,43.,7.,2.,0.,0.,2.,5.,5.,2.6,1.,1.,2.6,7.,44.,44.,43./	LM	14
15		HTH=HT/44.	LM	15
		CALL CALPLT (XO,YO,3)	LM	16
		DO 1 I=1,17	LM	17
		K=2	LM	18
		IF (I.EQ.1) K=3	LM	19
20		X=XO+(XM(I)+2.)*HTH	LM	20
		Y=YO+YM(I)*HTH	LM	21
	1	CALL CALPLT (X,Y,K)	LM	22
		CALL CALPLT (XO,YO,3)	LM	23
		RETURN	LM	24
25		END	LM	25-

CARD NO.

1		SUBROUTINE GRIDLN (XD,YD,XL,YH,NDIV)	GL	1
	C		GL	2
	C	ROUTINE TO DRAW AREA OF GRID LINES	GL	3
	C		GL	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	GL	5
	C		GL	6
	C	PARAMETER DEFINITION	GL	7
	C	XD,YD - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF GRID AREA	GL	8
	C	XL - LENGTH OF GRID AREA	GL	9
10	C	YH - HEIGHT OF GRID AREA	GL	10
	C	NDIV - NUMBER OF GRID LINES PER INCH	GL	11
	C		GL	12
	C	COMPUTE NUMBER OF GRID LINES	GL	13
		XD=FLOAT(NDIV)	GL	14
15		IF (XD.LE.0.0) XD=1.	GL	15
		NGX=IFIX(XD*XL+0.0001)+1	GL	16
		NGY=IFIX(XD*YH+0.0001)+1	GL	17
	C	DRAW GRID LINES PARALLEL TO Y-AXIS	GL	18
		CALL CALPLT (XD,YD,3)	GL	19
20		D=0.	GL	20
		IF (NGX.GT.1) D=XL/FLOAT(NGX-1)	GL	21
		X1=XD	GL	22
		Y1=YD	GL	23
		Y2=YD+YH	GL	24
25		L=0	GL	25
		DO 3 I=1,NGX	GL	26
		L=L+1	GL	27
		X2=X1+D	GL	28
		IF (L-1) 1,1,2	GL	29
30	1	CALL CALPLT (X1,Y2,2)	GL	30
		IF (I.EQ.NGX) GO TO 3	GL	31
		CALL CALPLT (X2,Y2,3)	GL	32
		GO TO 3	GL	33
	2	CALL CALPLT (X1,Y1,2)	GL	34
35		IF (I.EQ.NGX) GO TO 3	GL	35
		CALL CALPLT (X2,Y1,3)	GL	36
		L=0	GL	37
	3	X1=X2	GL	38
	C	DRAW GRID LINES PARALLEL TO X-AXIS	GL	39
40		CALL CALPLT (XD,YD,3)	GL	40

LISTING OF DECK: GRIDLN

PAGE 2

CARD NO.

41		D=0.	GL 41
		IF (NGY.GT.1) D=YH/FLOAT(NGY-1)	GL 42
		Y1=Y0	GL 43
		X1=X0	GL 44
45		X2=X0+XL	GL 45
		L=0	GL 46
		DO 6 I=1,NGY	GL 47
		L=L+1	GL 48
		Y2=Y1+D	GL 49
50		IF (L-1) 4,4,5	GL 50
	4	CALL CALPLT (X2,Y1,2)	GL 51
		IF (I.EQ.NGY) GO TO 6	GL 52
		CALL CALPLT (X2,Y2,3)	GL 53
		GO TO 6	GL 54
55	5	CALL CALPLT (X1,Y1,2)	GL 55
		IF (I.EQ.NGY) GO TO 6	GL 56
		CALL CALPLT (X1,Y2,3)	GL 57
		L=0	GL 58
	6	Y1=Y2	GL 59
60	C	FRAME GRID AREA	GL 60
		X1=X0+XL	GL 61
		Y1=Y0+YH	GL 62
		CALL CALPLT (X0,Y0,3)	GL 63
		CALL CALPLT (X1,Y0,2)	GL 64
65		CALL CALPLT (X1,Y1,2)	GL 65
		CALL CALPLT (X0,Y1,2)	GL 66
		CALL CALPLT (X0,Y0,2)	GL 67
		CALL CALPLT (X0,Y0,3)	GL 68
	C	RETURN TO CALLING PROGRAM	GL 69
70		RETURN	GL 70
		END	GL 71-

CARD NO.

1		SUBROUTINE RUNKEY (X0,Y0,RUN,LSYM,ISIZE,HT,NRNMAX)	KY	1
	C		KY	2
	C	ROUTINE TO DRAW TITLE BLOCK CONTAINING SYMBOLS AND RUN NUMBERS	KY	3
	C		KY	4
5	C	CODED BY -- HARRY L. MORGAN NASA/LARC/TAD/AAB 1983	KY	5
	C		KY	6
	C	PARAMETER DEFINITION	KY	7
	C	X0,Y0 - X,Y COORDINATES OF LOWER LEFT-HAND CORNER OF KEY	KY	8
	C	RUN - RUN NUMBERS	KY	9
10	C	LSYM - SYMBOL ORDER	KY	10
	C	ISIZE - SYMBOL SIZE	KY	11
	C	HT - SYMBOL HEIGHT	KY	12
	C	NRNMAX - MAXIMUM NUMBER OF ALLOWABLE RUNS PER SHEET	KY	13
	C		KY	14
15	C	DIMENSION RUN(1), LSYM(1)	KY	15
	C	WORK ARRAYS	KY	16
	C	COMMON /PLT/ KRUN(10),NRUN(10),LRUN(22),JSYM(22)	KY	17
	C		KY	18
	C	INTEGER RUN	KY	19
20	C	INITIALIZE ROUTINE CONSTANTS	KY	20
	C	DATA XM/1.0/,XMARG/0.1/,XNSP/2.5/,SPC/0.2/	KY	21
	C	DEFINE LETTERING SIZE	KY	22
	C	SZ=HT*XM	KY	23
	C	INITIALIZE CHARACTER SET	KY	24
25	C	CALL CHARST2	KY	25
	C		KY	26
	C	DETERMINE THE NUMBER OF NON-ZERO RUNS AND CORRESPONDING SYMBOLS	KY	27
	C		KY	28
	C	NR=0	KY	29
30		DO 2 I=1,NRNMAX	KY	30
		IF (RUN(I)) 1,2,1	KY	31
	1	NR=NR+1	KY	32
		LRUN(NR)=RUN(I)	KY	33
		JSYM(NR)=LSYM(I)	KY	34
35	2	CONTINUE	KY	35
		IF (NR.EQ.0) RETURN	KY	36
	C		KY	37
	C	COMPUTE WIDTH OF WORDS SYMBOL AND 99999	KY	38
	C		KY	39
40		CALL CHARWH (SYMW,HA,HBS,SZ,BH)S(YMBOL,8,SPC)	KY	40

CARD NO.

41		CALL CHARWH (WD,HA,HB,SZ,5H99999,5,SPC)	KY	41
		XS=XO+0.5*SYMW+XMARG	KY	42
		XR=XO+2.*SYMW+0.5*WD+XMARG	KY	43
		YSR=Y0+XMARG	KY	44
45	C		KY	45
	C	DRAW SYMBOL AND RUN NUMBER	KY	46
	C		KY	47
		DO 6 L=1,NR	KY	48
		I=NR+1-L	KY	49
50		ENCODE (10,7,MRUN) LRUN(I)	KY	50
		DECODE (10,8,MRUN) KRUN	KY	51
	C	FIND NUMBER OF DIGITS IN RUN NUMBER	KY	52
		DO 3 J=1,10	KY	53
	3	NRUN(J)=1H	KY	54
55		KDIG=0	KY	55
		DO 4 J=1,10	KY	56
		IF (KRUN(J).EQ.1H) GO TO 4	KY	57
		KDIG=KDIG+1	KY	58
	C	SHIFT RUN NUMBER FROM RIGHT TO LEFT JUSTIFIED	KY	59
60		NRUN(KDIG)=KRUN(J)	KY	60
	4	CONTINUE	KY	61
	C	REFORMAT RUN NUMBER	KY	62
		ENCODE (10,8,MRUN) NRUN	KY	63
		DECODE (10,9,MRUN) IRUN	KY	64
65	C	DRAW SYMBOL	KY	65
		Y2=YSR+0.5*SZ	KY	66
		CALL PNTPLT (XS,Y2,JSYM(I),ISIZE)	KY	67
	C	DRAW RUN NUMBER	KY	68
		IF (KDIG.EQ.0) GO TO 5	KY	69
70		CALL CHARWH (WD,HA,HB,SZ,IRUN,KDIG,SPC)	KY	70
		XX=XR-0.5*WD	KY	71
		CALL CHARACT (XX,YSR,SZ,IRUN,0.,KDIG,SPC)	KY	72
	5	YSR=YSR+XNSP*SZ	KY	73
	6	CONTINUE	KY	74
75	C		KY	75
	C	DRAW AND UNDERLINE WORD SYMBOL AND RUN	KY	76
	C		KY	77
		CALL CHARWH (WD,HA,HB,SZ,5H)R(UN,5,SPC)	KY	78
		IF (HB.LT.HBS) HBS=HB	KY	79
80		XX=XS-0.5*(SYMW+XMARG)	KY	80

CARD NO.

81		CALL CALPLT (XX,YSR,3)	KY 81
		XX=XS+0.5*(SYMW+XMARG)	KY 82
		CALL CALPLT (XX,YSR,2)	KY 83
		XX=XR-0.5*(WD+XMARG)	KY 84
85		CALL CALPLT (XX,YSR,3)	KY 85
		XX=XR+0.5*(WD+XMARG)	KY 86
		CALL CALPLT (XX,YSR,2)	KY 87
		YSR=YSR-HBS+0.5*XMARG	KY 88
		XX=XS-0.5*SYMW	KY 89
90		CALL CALPLT (XX,YSR,3)	KY 90
		CALL CHARACT (XX,YSR,SZ,8H)S(YMBOL,0.,8,SPC)	KY 91
		XX=XR-0.5*WD	KY 92
		CALL CHARACT (XX,YSR,SZ,5H)R(UN,0.,5,SPC)	KY 93
	C		KY 94
95	C	RETURN TO TITLE BLOCK ORIGIN	KY 95
	C		KY 96
		CALL CALPLT (XD,YD,3)	KY 97
		CALL CHARST1	KY 98
	C		KY 99
100		RETURN	KY 100
	C		KY 101
	7	FORMAT (I10)	KY 102
	8	FORMAT (10A1)	KY 103
	9	FORMAT (A10)	KY 104
105		END	KY 105-

APPENDIX B

DESCRIPTION OF INPUT DATA FOR WIND-TUNNEL DATA

PLOTING PROGRAM PLOTWD

This appendix contains a description of the input requirements for the wind-tunnel data plotting program PLOTWD. The input data is divided into two basic parts: (1) the plotting setup information and (2) the plotting sheet information. The setup information defines the data variables, axis scales and positions, and the type of data fairing desired. All setup variables are input as floating point quantities with a format of F10.0, except the YLABEL and XLABEL variables which are axis labels and are input with a format of A10.

Plotting Setup

<u>CARD</u>	<u>VARIABLE</u>	<u>VALUE</u>	<u>DESCRIPTION</u>
1	TEST	-	Test number
2	SHEETW	-	Total width of plotting sheet, in.
	SHEETH	-	Total height of plotting sheet, in.
	SPACE	-	Space between plotting sheets, in.
3	ISYM	-	Starting symbol number (See table I.)
	ISIZE	1	Small symbol size
		2	Medium symbol size
		3	Large symbol size
4	IOP	0	Plot data points only
		1	Plot data points and fair with tension spline
	TENSION	-	Tension factor, $\bar{\sigma}$
5	IEDIT	0	Do not sort or edit data
		1	Sort and edit data
	TOLR	-	Edit tolerance of independent test variable (TOLR=0 for sort only)
6	IW	-	Data array location of independent test variable

<u>CARD</u>	<u>VARIABLE</u>	<u>VALUE</u>	<u>DESCRIPTION</u>
7	NPLOT	-	Number of plots per sheet
8	YTAPE	-	Data array location of y-axis variable
	YOFFSET	-	Sheet height location of y-axis origin, in.
	YSCALE	-	Scale value per inch for y-axis
	YLABEL	-	Label for y-axis
	XTAPE	-	Data array location of x-axis variable
	XOFFSET	-	Sheet width location of x-axis origin, in.
	XSCALE	-	Scale value per inch for x-axis
	XLABEL	-	Label for x-axis

Card 8 is input NPLOT times and NPLOT is limited to a value of 10 for each setup.

Plotting Sheet Namelist

<u>NAMELIST VARIABLE</u>	<u>VALUE</u>	<u>DESCRIPTION</u>
\$SHEET	-	Namelist label
NO	-	Sheet number
RUN	-	Run numbers to be plotted on sheet
NEWCASE	0 or default	A new SHEET namelist follows
	1	A new setup deck follows
\$END	-	End of namelist

Note that no part of the namelist may be in column 1 and that a maximum of 10 run numbers may be input per sheet.

APPENDIX C

DESCRIPTION OF OUTPUT FOR WIND-TUNNEL DATA PLOTTING PROGRAM PLOTWD

This appendix contains a description of the typical output for the wind-tunnel data plotting program PLOTWD. A sample four-page output is presented in table IV and was generated from the upper set of sample input data presented in table III. Page 1 of the output is a printout of the input setup data as described in appendix B. This page should be carefully checked after each program execution to insure that the desired plotting variables and scales have been properly input. Page 2 of the output is a list of the run numbers contained in the data copied from the input file TAPE1 to the random-access file TAPE2. The total number of data points copied to the random-access file is printed following the list of run numbers. The quantity NPMAX and the dimensioned size of the array NPT in the main program PLOT should be equal to or greater than the total number of data points copied to file TAPE2.

Page 3 of the output is a list of the sheet and corresponding run numbers plotted. A run number with the value zero indicates that no data were plotted with the symbol corresponding to the order of the run number in the list. If the user-specified run number is not available on the random-access file, a message will be printed stating that the specified run could not be found. In addition, if during the plotting of the scaled data any data points fall outside the sheet boundaries, a message will be printed stating the number of data points outside the boundaries. These outside data points will not be plotted and, therefore, may require that the user redefine either the axis scale factors or the sheet height and width.

If the user selects the sort and edit option, page 4 will be output which contains a summary of the sort and edit information for each run called during the program execution. The first set of values listed in this summary are the input values of the independent variable t for the particular run. The sorted

t values listed next represent the results of the sorting procedure which simply reorders the input values in a monotonically increasing order. The sorted t values for a particular run will be the same regardless of the number of times the run is plotted. The edited and deleted t values listed last represent the results of the editing procedure which deletes all but one of the multiple data points within the specified tolerance TOLR. The data points deleted are selected based on the input y values for the particular part of the figure and, therefore, may be different for each part. The edited and deleted t values listed are only for the first part of the figure. The deleted points are not used during the spline curve fairing process, but nevertheless, they are plotted with the appropriate symbol.

If the user replaces the call to subroutine CURPLT in the main program PLOT with a call to subroutine LSQPLT, the sample output summary information presented in table V will be listed for the least-squares polynomial curve fit option and in table VI for the least-squares cubic-spline curve fit option. The parameter IPRINT in subroutine LSQPLT must also be set equal to 1 for the summary information to be listed. This information will appear in the output immediately following the print of the particular sheet and corresponding run numbers on page 3 of the basic output. The summary information will be listed each time the subroutine is called; therefore, the user is cautioned that a large amount of printed output can easily be generated even if a moderate number of plots and runs are specified.

The summary information presented in tables V and VI consists of a tabulated listing of the input t, x, and y values, the new x and y values generated by the least-squares curve fit, and the differences between input and new x and y values. Immediately following the tabulated listing, the sum-of-the-squares of the differences between the input and new x and y values are listed as an error-in-x and -y, respectively. If the independent

variable t and the x -axis variable are the same as it is for the sample output case, the error-in- x will be very small. If the least-square polynomial curve fit option is chosen, the error-in- x and $-y$ will be a function of the degree of the polynomial. If the least-square cubic-spline option is chosen, the error-in- x and $-y$ will always be equal to the product of the number of input points times the square of the standard deviation DF .

TABLE I.- LIST OF STANDARD SYMBOL AND DASHED LINE COMBINATIONS

ISYM	SYMBOL	DASHED LINE
1	○	_____
2	□	-----
3	◇	-----
4	△	-----
5	▵	-----
6	▹	-----
7	◐	-----
8	◑	-----
9	◒	-----
10	◓	-----
11	⊕	-----
12	⊞	-----
13	⊠	-----
14	⊡	-----
15	⊢	-----
16	⊣	-----
17	⊤	-----
18	⊥	-----
19	⊦	-----
20	⊧	-----
21	•	-----
22	+	-----

TABLE II. - LIST OF STANDARD AERODYNAMIC SYMBOLS DRAWN BY SUBROUTINE COEFSY

ISYM	COEFFICIENT	SYMBOL	WIDTH/HEIGHT
1	Lift	C_L	2.0
2	Drag	C_D	2.1
3	Pitching Moment	C_m	2.2
4	Rolling Moment	C_l	1.7
5	Yawing Moment	C_n	1.9
6	Side Force	C_Y	2.1
7	Lift-Drag Ratio	L/D	2.6
8	Angle of Attack	α, deg	3.5
9	Angle of Sideslip	β, deg	3.4

TABLE IV. - SAMPLE OUTPUT FOR CASE 1 INPUT

PAGE 1 OUTPUT									
PLOTTING SETUP DECK FOR TEST = 496									
SHEET WIDTH = 20.00		SHEET HEIGHT = 23.00		SPACE BETWEEN SHEETS = 10.00					
STARTING SYMBOL NUMBER = 1		SYMBOL SIZE IS LARGE							
DATA WILL BE FAIRED AND PLOTTED USING SPLINE WITH TENSION = 0.00									
DATA WILL BE EDITED WITH AN EDIT TOLERANCE = .500									
TAPE LOCATION OF INDEPENDENT VARIABLE = 1									
NUMBER OF PLOTS PER SHEET = 3									
PLOT NO.	YTAPE	YOFFSET	YSCALE	YLABEL	XTAPE	XOFFSET	XSCALE	XLABEL	
1	6	2.00	.4000	CL	1	4.00	2.0000	ALPHA	
2	7	11.00	.2000	CD	1	4.00	2.0000	ALPHA	
3	8	20.00	.4000	CM	1	4.00	2.0000	ALPHA	

PAGE 2 OUTPUT									
THE FOLLOWING LIST OF RUNS ARE AVAILABLE ON THE DATA DISK FOR TEST NO. 496									
2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	21
22	23	24	25	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42
43	44	45	46	47	48	49	50	51	52
53	54	55	56	57	58	59	60	61	62
63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82
83	84	85	86	87	88	89	90	91	92
93	94	95	96	97	98	99	100	101	102
103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122
123	124	125	126	127	128	129	130	131	132
133	134	135	136	137	138	139	140	141	142
143	144	145	146	147	148	149	150	151	152
153	154	155	156	157	158	159	160	161	162
163	164	165	166	167	168	169	170	171	172
173	174	175	176	177	178	179	180	181	182
183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202
203	204	205	206	207	208	209	210	211	212
213	214	215	216	217	218	219	220	221	222
223	224	225	226	227	228	229	230	231	232
233	234	235	236	237	238	239	240	241	242
243	244	245	246	247	248	249	250	251	252
253	254								
TOTAL NUMBER OF DATA POINTS = 4901									

PAGE 3 OUTPUT									
DATA PLOTTED									
SHEET NO = 206	RUNS =	39,	59,	0,	0,	0,	0,	0,	0

TABLE IV. - CONCLUDED

PAGE 4 OUTPUT

THE FOLLOWING RUNS HAVE BEEN EDITED

RUN NO. = 39 TOLERANCE = .500

INPUT		SORTED		EDITED		DELETED	
POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE
1	-6.1700	1	-6.1700	1	-6.1700	4	.3500
2	-4.1800	2	-4.1800	2	-4.1800		
3	-1.8800	3	-1.8800	3	-1.8800		
4	.3500	20	.1700	20	.1700		
5	2.0100	4	.3500	5	2.0100		
6	4.2800	5	2.0100	6	4.2800		
7	6.3700	6	4.2800	7	6.3700		
8	8.4500	7	6.3700	8	8.4500		
9	10.3200	8	8.4500	9	10.3200		
10	12.4100	9	10.3200	10	12.4100		
11	14.4000	10	12.4100	11	14.4000		
12	16.8000	11	14.4000	12	16.8000		
13	18.6700	12	16.8000	13	18.6700		
14	20.8600	13	18.6700	14	20.8600		
15	22.7500	14	20.8600	15	22.7500		
16	24.7800	15	22.7500	16	24.7800		
17	26.6700	16	24.7800	17	26.6700		
18	28.7500	17	26.6700	18	28.7500		
19	30.7000	18	28.7500	19	30.7000		
20	.1700	19	30.7000				

RUN NO. = 59 TOLERANCE = .500

INPUT		SORTED		EDITED		DELETED	
POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE	POINT NO.	T-VALUE
1	-6.0200	1	-6.0200	1	-6.0200	4	.2800
2	-4.4000	2	-4.4000	2	-4.4000		
3	-2.2200	3	-2.2200	3	-2.2200		
4	.2800	4	.2800	20	.2900		
5	2.0000	20	.2900	5	2.0000		
6	4.8700	5	2.0000	6	4.8700		
7	6.3000	6	4.8700	7	6.3000		
8	8.2000	7	6.3000	8	8.2000		
9	10.5800	8	8.2000	9	10.5800		
10	12.6400	9	10.5800	10	12.6400		
11	14.9700	10	12.6400	11	14.9700		
12	16.6100	11	14.9700	12	16.6100		
13	18.4600	12	16.6100	13	18.4600		
14	20.8900	13	18.4600	14	20.8900		
15	22.8300	14	20.8900	15	22.8300		
16	25.3700	15	22.8300	16	25.3700		
17	26.9800	16	25.3700	17	26.9800		
18	28.9200	17	26.9800	18	28.9200		
19	31.3400	18	28.9200	19	31.3400		
20	.2900	19	31.3400				

ALL PLOTTING COMPLETED

TABLE V.- SAMPLE OF LEAST-SQUARES POLYNOMIAL CURVE FIT

DATA PLOTTED

SHEET NO = 206 RUNS = 39, 59, 0, 0, 0, 0, 0, 0,

RESULTS OF LEAST SQUARES POLYNOMIAL CURVE FIT FOR RUN = 39 NSYM = 1 NPOL = 3

I	T	X	XNEW	X-XNEW	Y	YNEW	Y-YNEW
1	-6.1700	.9150	.9150	-.0000	1.1329	1.0588	.0741
2	-4.1800	1.9100	1.9100	.0000	1.4455	1.4068	.0387
3	-1.8800	3.0600	3.0600	.0000	1.7651	1.8479	-.0828
4	.1700	4.0850	4.0850	.0000	2.2144	2.2678	-.0534
5	2.0100	5.0050	5.0050	.0000	2.6249	2.6597	-.0348
6	4.2800	6.1400	6.1400	.0000	3.1164	3.1541	-.0377
7	6.3700	7.1850	7.1850	.0000	3.5735	3.6117	-.0382
8	8.4500	8.2250	8.2250	.0000	4.0115	4.0610	-.0495
9	10.3200	9.1600	9.1600	.0000	4.4649	4.4533	.0116
10	12.4100	10.2050	10.2050	0.0000	4.9459	4.8711	.0747
11	14.4000	11.2000	11.2000	-.0000	5.4040	5.2417	.1623
12	16.8000	12.4000	12.4000	-.0000	5.8305	5.6430	.1874
13	18.6700	13.3350	13.3350	-.0000	5.9419	5.9138	.0281
14	20.8600	14.4300	14.4300	-.0000	6.0995	6.1757	-.0763
15	22.7500	15.3750	15.3750	-.0000	6.2170	6.3472	-.1301
16	24.7800	16.3900	16.3900	-.0000	6.3325	6.4677	-.1352
17	26.6700	17.3350	17.3350	.0000	6.4900	6.5143	-.0243
18	28.7500	18.3750	18.3750	.0000	6.3650	6.4852	-.1202
19	30.7000	19.3500	19.3500	.0000	6.5800	6.3744	.2056

ERROR IN X = .154875E-22

ERROR IN Y = .189547E+00

TABLE VI. - SAMPLE OF LEAST-SQUARES CUBIC-SPLINE CURVE FIT

DATA PLOTTED										
SHEET NO = 206		RUNS = 39,		59,	0,	0,	0,	0,	0,	0,
RESULTS OF LEAST SQUARES CUBIC SPLINE CURVE FIT FOR RUN = 39								NSYM = 1	DF = .10000	
I	T	X	XNEW	X-XNEW	Y	YNEW	Y-YNEW			
1	-6.1700	.9150	.9150	0.0000	1.1329	1.0207	.1122			
2	-4.1800	1.9100	1.9100	.0000	1.4455	1.4164	.0291			
3	-1.8800	3.0600	3.0600	0.0000	1.7651	1.8793	-.1142			
4	.1700	4.0850	4.0850	0.0000	2.2144	2.3021	-.0877			
5	2.0100	5.0050	5.0050	.0000	2.6249	2.6911	-.0662			
6	4.2800	6.1400	6.1400	.0000	3.1164	3.1807	-.0643			
7	6.3700	7.1850	7.1850	.0000	3.5735	3.6354	-.0619			
8	8.4500	8.2250	8.2250	0.0000	4.0115	4.0837	-.0722			
9	10.3200	9.1600	9.1600	0.0000	4.4649	4.4744	-.0095			
10	12.4100	10.2050	10.2050	0.0000	4.9459	4.8855	.0604			
11	14.4000	11.2000	11.2000	.0000	5.4040	5.2396	.1644			
12	16.8000	12.4000	12.4000	.0000	5.8305	5.6055	.2250			
13	18.6700	13.3350	13.3350	.0000	5.9419	5.8400	.1018			
14	20.8600	14.4300	14.4300	.0000	6.0995	6.0624	.0371			
15	22.7500	15.3750	15.3750	.0000	6.2170	6.2162	.0009			
16	24.7800	16.3900	16.3900	.0000	6.3325	6.3511	-.0186			
17	26.6700	17.3350	17.3350	0.0000	6.4900	6.4569	.0331			
18	28.7500	18.3750	18.3750	.0000	6.3650	6.5606	-.1956			
19	30.7000	19.3500	19.3500	.0000	6.5800	6.6538	-.0738			
ERROR IN X = .575553E-25				ERROR IN Y = .190000E+00						

TABLE VII. - COMPUTER CODE FOR CASE 1

```

C      DEFINE HEIGHT OF COEFFICIENTS
      HT=0.4
C      DEFINE HEIGHT OF AXIS LABELS
      SZ=0.75*HT
C      PROVIDE A BLANK SPACE AT BOTTOM OF PLOT FOR FIGURE TITLE
      CALL CALPLT (0.0,4.0,-3)
C      DRAW GRID LINES
      CALL GRIDLN (0.0,0.0,20.0,23.0,1)
C      DRAW ZERO LINES
      CALL ZEROLN (4.0,0.0,23.0,2)
      CALL ZEROLN (0.0,2.0,20.0,1)
      CALL ZEROLN (0.0,11.0,20.0,1)
      CALL ZEROLN (0.0,20.0,20.0,1)
C      DRAW AND LABEL X-AXIS ( ALPHA )
      CALL AXISLB (1,0.0,0.0,-8.0,2.0,20.0,SZ,-1)
      XD=(20.0-3.5*HT)/2.0
      YD=-1.0
      CALL COEFSY (XD,YD,HT,8)
C      DRAW AND LABEL LOWER PORTION OF Y-AXIS ( CL )
      CALL AXISLB (2,0.0,0.0,-0.8,0.4,10.0,SZ,1)
      XD=-1.8
      YD=5.0
      CALL COEFSY (XD,YD,HT,1)
C      DRAW AND LABEL MIDDLE PORTION OF Y-AXIS ( CD )
      CALL AXISLB (2,0.0,11.0,0.0,0.2,5.0,SZ,1)
      XD=-1.8
      YD=13.5
      CALL COEFSY (XD,YD,HT,2)
C      DRAW AND LABEL UPPER PORTION OF Y-AXIS ( CM )
      CALL AXISLB (2,0.0,17.0,-1.2,0.4,6.0,SZ,1)
      XD=-1.8
      YD=20.0
      CALL COEFSY (XD,YD,HT,3)
C      DRAW AND LABEL RUN KEY
      XD=2.0
      YD=HT+23.0
      CALL RUNKEY (XD,YD,RUN,LSYM,ISIZE,HT,10)

```

TABLE VIII.- COMPUTER CODE FOR CASE 2

```

C   DEFINE HEIGHT OF COEFFICIENTS
    HT=0.25
C   DEFINE HEIGHT OF AXIS LABELS
    SZ=0.75*HT
C   PROVIDE A BLANK SPACE AT BOTTOM OF PLOT FOR FIGURE TITLE
    CALL CALPLT (0.0,3.0,-3)

C
C   LOWER LEFT-HAND PLOT   ( CM VS CL )
C
C       DRAW GRID LINES AND ZERO LINES
    CALL GRIDLN (0.0,0.0,5.0,5.0,1)
    CALL ZEROLN (1.0,0.0,5.0,2)
    CALL ZEROLN (0.0,2.0,5.0,1)
C       DRAW AND LABEL X-AXIS
    CALL AXISLB (1,0.0,0.0,-0.8,0.8,5.0,SZ,1)
    XO=(5.0-2.0*HT)/2.0
    YO=-0.8
    CALL COEFSY (XO,YO,HT,1)
C       DRAW AND LABEL Y-AXIS
    CALL AXISLB (2,0.0,0.0,-0.8,0.4,5.0,SZ,1)
    XO=-1.0
    YO=2.5
    CALL COEFSY (XO,YO,HT,3)

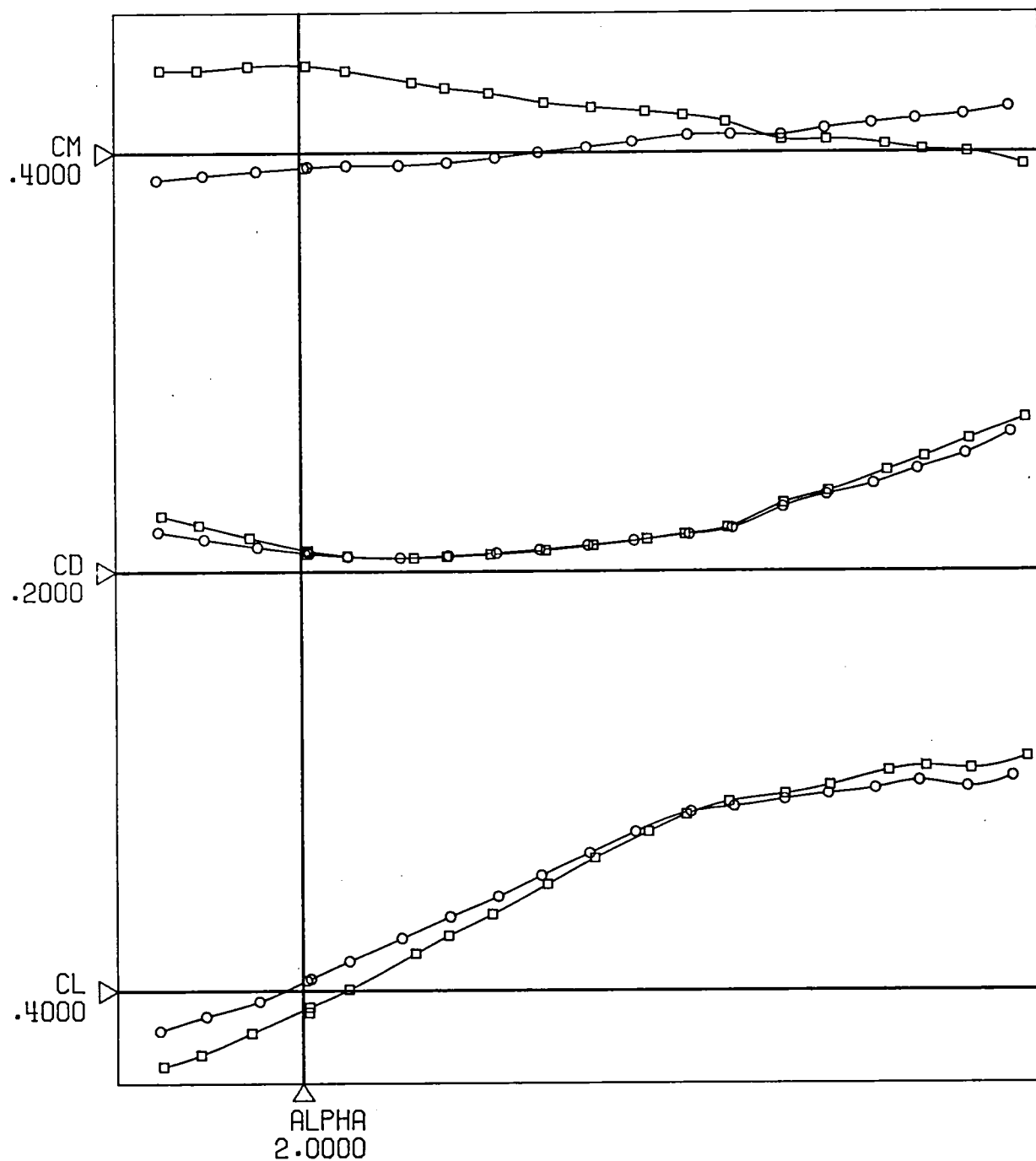
C
C   LOWER RIGHT-HAND PLOT ( L/D VS ALPHA )
C
C       DRAW GRID LINES AND ZERO LINES
    CALL GRIDLN (7.0,0.0,5.0,5.0,1)
    CALL ZEROLN (8.0,0.0,5.0,2)
    CALL ZEROLN (7.0,1.0,5.0,1)
C       DRAW AND LABEL X-AXIS
    CALL AXISLB (1,7.0,0.0,-10.0,10.0,5.0,SZ,-1)
    XO=7.0+(5.0-3.5*HT)/2.0
    YO=-0.8
    CALL COEFSY (XO,YO,HT,8)
C       DRAW AND LABEL Y-AXIS
    CALL AXISLB (2,7.0,0.0,-4.0,4.0,5.0,SZ,-1)
    XO=6.0
    YO=2.5
    CALL COEFSY (XO,YO,HT,7)

C
C   UPPER LEFT-HAND PLOT   ( CL VS ALPHA )
C
C       DRAW GRID LINES AND ZERO LINES
    CALL GRIDLN (0.0,7.0,5.0,5.0,1)
    CALL ZEROLN (1.0,7.0,5.0,2)
    CALL ZEROLN (0.0,8.0,5.0,1)
C       DRAW AND LABEL X-AXIS
    CALL AXISLB (1,0.0,7.0,-10.0,10.0,5.0,SZ,-1)
    XO=(5.0-3.5*HT)/2.0
    YO=6.2
    CALL COEFSY (XO,YO,HT,8)
C       DRAW AND LABEL Y-AXIS
    CALL AXISLB (2,0.0,7.0,-0.8,0.8,5.0,SZ,1)
    XO=-1.0
    YO=9.5
    CALL COEFSY (XO,YO,HT,1)

C
C   UPPER RIGHT-HAND PLOT ( CD VS CL )
C
C       DRAW GRID LINES AND ZERO LINES
    CALL GRIDLN (7.0,7.0,5.0,5.0,1)
    CALL ZEROLN (8.0,7.0,5.0,2)
    CALL ZEROLN (7.0,7.0,5.0,1)
C       DRAW AND LABEL X-AXIS
    CALL AXISLB (1,7.0,7.0,-0.8,0.8,5.0,SZ,1)
    XO=7.0+(5.0-2.0*HT)/2.0
    YO=6.2
    CALL COEFSY (XO,YO,HT,1)
C       DRAW AND LABEL Y-AXIS
    CALL AXISLB (2,7.0,7.0,0.0,0.2,5.0,SZ,1)
    XO=6.0
    YO=9.5
    CALL COEFSY (XO,YO,HT,2)

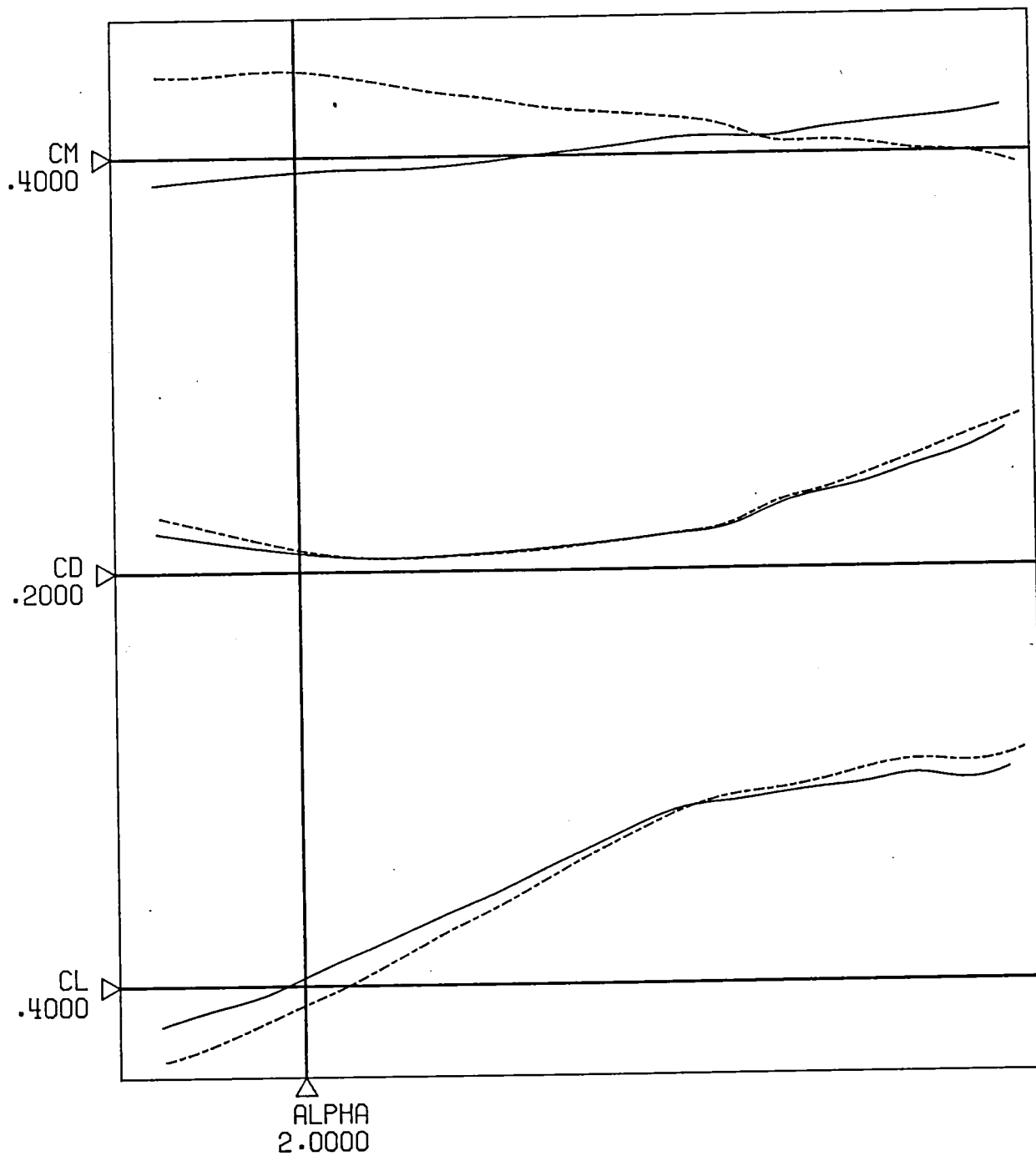
C
C   DRAW AND LABEL RUN KEY
C
    XO=1.0
    YO=12.0+HT
    CALL RUNKEY (XO,YO,RUN,LSYM,ISIZE,HT,10)

```



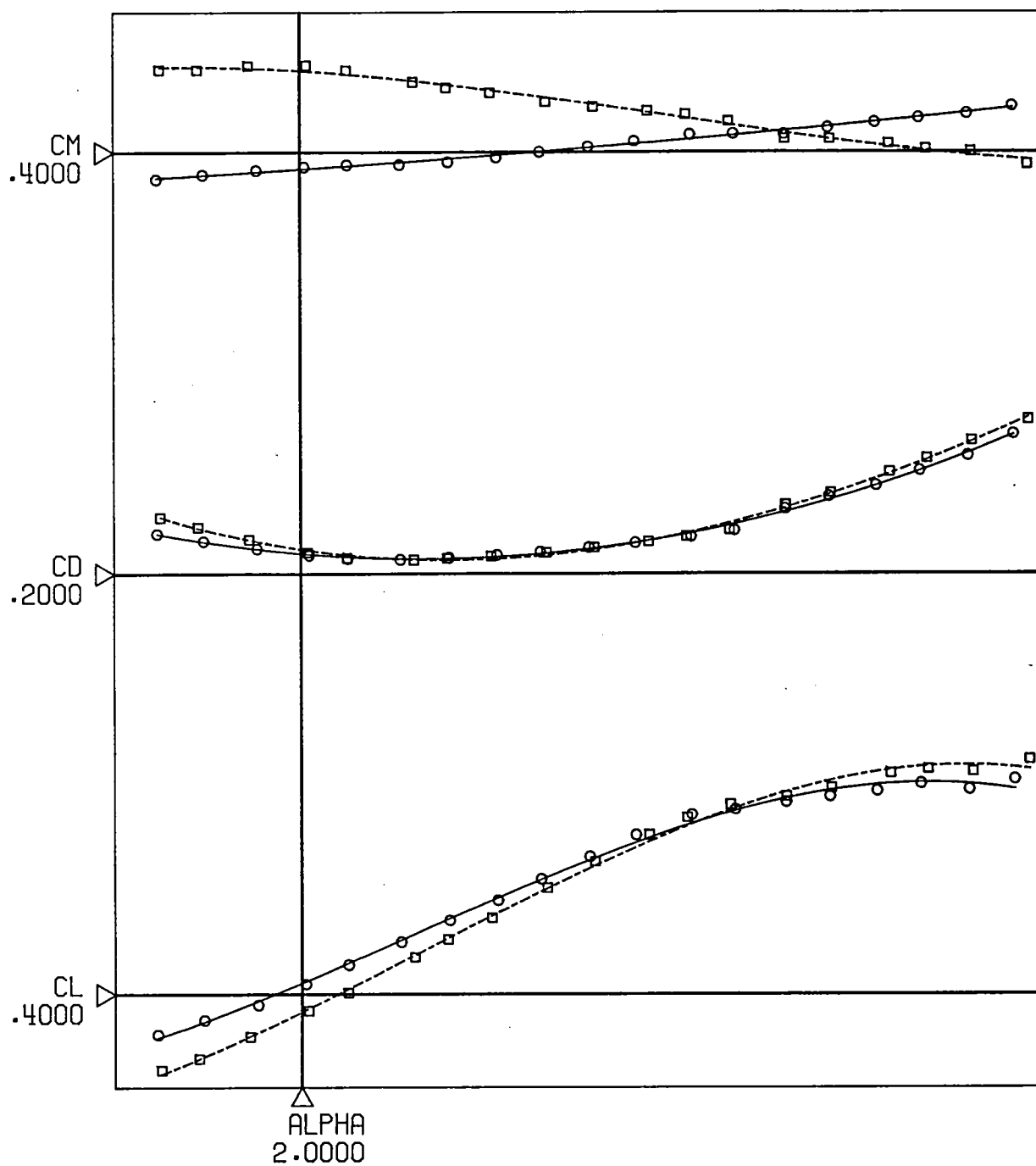
TEST 496 ,SHEET 206 ,RUNS 39 59 0 0 0 0 0

Figure 1. - Plot of Case 1 input with standard symbols and cubic-spline curve fairing.



TEST 496 ,SHEET 206 ,RUNS 39 59 0 0 0 0 0

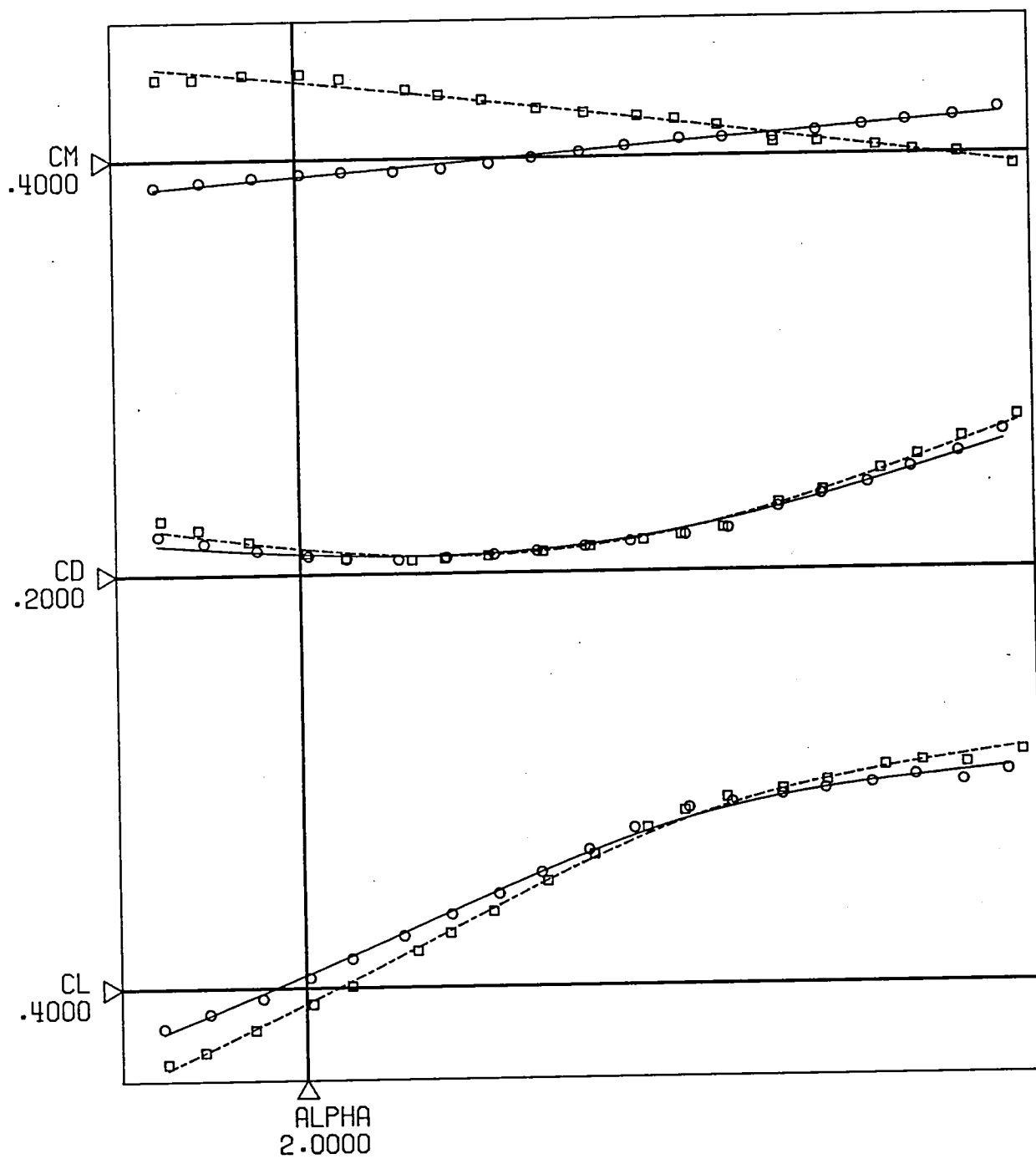
Figure 2. - Plot of Case 1 input with standard dashed line combination and cubic-spline curve fairing.



TEST 496 ,SHEET 206 ,RUNS 39 59 0 0 0. 0. 0

a) Least-Squares Polynomial

Figure 3. - Plot of Case 1 input with least-squares curve fairing.



TEST 496 , SHEET 206 , RUNS 39 59 0 0 0 0 0

b) Least-Squares Cubic Spline

Figure 3. - Concluded.

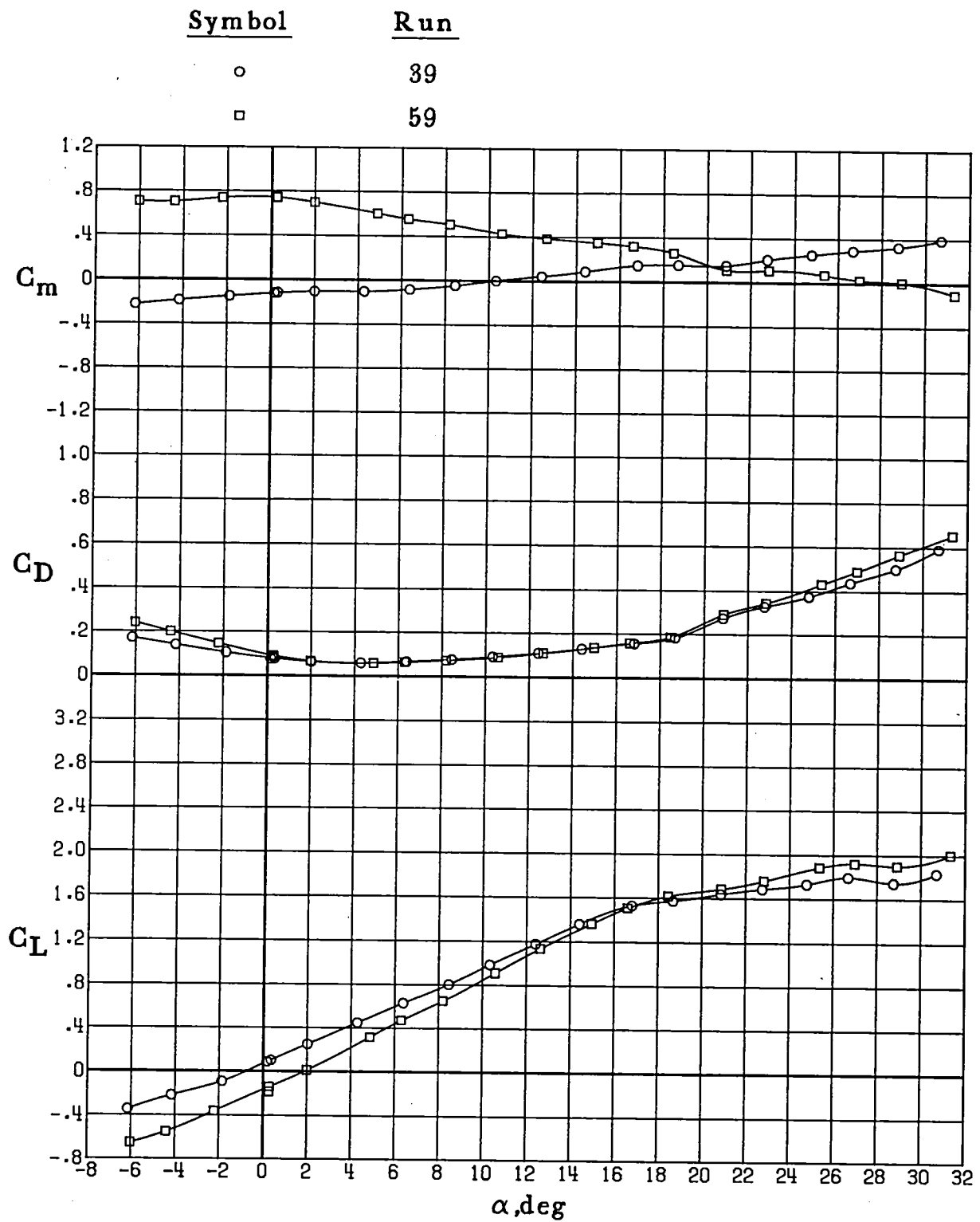


Figure 4. - Plot of Case 1 input with grid lines, axis scale values and labels, and legend drawn.

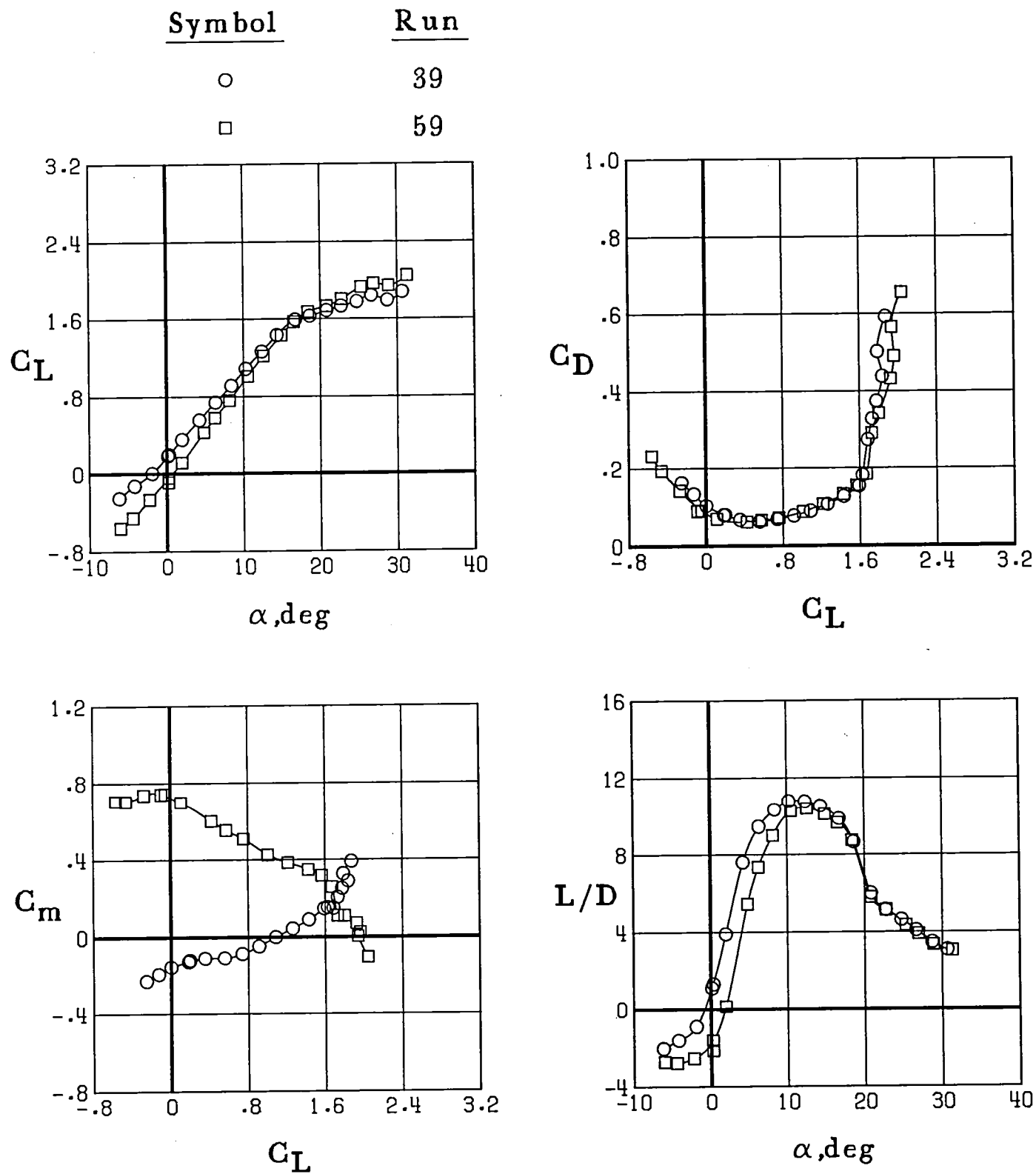


Figure 5.- Plot of Case 2 input with grid lines, axis scale values and labels, and legend drawn.

1. Report No. NASA TM-85648		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Computer Program for Plotting and Fairing Wind-Tunnel Data				5. Report Date July 1983	
				6. Performing Organization Code 505-31-23-06	
7. Author(s) Harry L. Morgan, Jr.				8. Performing Organization Report No.	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665				10. Work Unit No.	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546				13. Type of Report and Period Covered Technical Memorandum	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract <p>This report contains a detailed description of the Langley computer program PLOTWD which plots and fairs experimental wind-tunnel data. The program was written for use primarily on the Langley CDC computer and CALCOMP plotters. The fundamental operating features of the program are that the input data are read and written to a random-access file for use during program execution, that the data for a selected run can be sorted and edited to delete duplicate points, and that the data can be plotted and faired using tension splines, least-squares polynomial, or least-squares cubic-spline curves. The most noteworthy feature of the program is the simplicity of the user-supplied input requirements. Several subroutines are also included that can be used to draw grid lines, zero lines, axis scale values and labels, and legends. A detailed description of the program operational features and each sub-program are presented. The general application of the program is also discussed together with the input and output for two typical plot types. A listing of the program code, user-guide, and output description are presented in appendices. The program has been in use at Langley for several years and has proven to be both easy to use and versatile.</p>					
17. Key Words (Suggested by Author(s)) Plotting and fairing Wind-tunnel data Computer program Tension splines			18. Distribution Statement RESTRICTED Subject Category 02		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 131	
22. Price*					

DO NOT REMOVE SLIP FROM MATERIAL	
Delete your name from this slip when returning material to the library.	
NAME	MS
Jim BYRD	413